

LIS User's Guide

July 24, 2009

Revision 6.0

History:

Revision	Summary of Changes	Date
6.0	LIS version 6.0 release	November 1, 2008



National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, Maryland 20771

History:

Revision	Summary of Changes	Date
5.0	Data Assimilation capability	September 11, 2006
4.2	AGRMET supported version	May 11, 2006
4.1	LIS version 4.0.2 release	March 14, 2005
4.0	Milestone "K" submission	February 11, 2005
3.2	LIS version 3.1 release	December 17, 2004
3.1	Milestone "G" release	July 16, 2004
3.0	Milestone "G" submission	May 7, 2004
2.3	Improvements to Milestone "T"	November 30, 2003
2.2	-	-
2.1	Milestone "I" release	November 10, 2003
2.0	Milestone "I" submission	August 14, 2003
1.1	Milestone "F" release	April 25, 2003
1.0	Milestone "F" submission	March 31, 2003

Contents

1	Introduction	7
1.1	What's New	7
1.1.1	LIS 5.0-6.0	7
1.1.2	LIS 4.2-5.0	8
1.1.3	LIS 4.1-4.2	8
1.1.4	LIS 4.0.2 – 4.1	9
1.1.5	LIS 4.0 – 4.0.2	9
1.1.6	LIS 3.1 – 4.0	9
1.1.7	LIS 3.0 – 3.1	9
1.1.8	LIS 2.0 – 3.0	10
2	Background	11
2.1	LIS	11
2.2	LIS core	12
3	Preliminaries	14
4	Obtaining the Source Code	15
4.1	Downloading the Source Code	15
4.2	Source files	15
5	Building the Executable	22
5.0.1	Development Tools	22
5.0.2	Required Software Libraries	22
5.0.3	Optional Software Libraries	22
5.0.4	Build Instructions	23
5.1	Generating documentation	25
6	Running the Executable	26
7	Test Cases	27
7.1	The <i>testcases</i> Sub-directory	27
7.2	Test Cases Input	27
7.3	Test Cases Output	27
7.3.1	Output Example	28
8	Output Data Processing	30
8.1	Binary output format	30
8.2	GRIB1 output format	31
8.3	NetCDF output format	31

9 LIS config File	32
9.1 Overall driver options	32
9.2 Parameter options	37
9.3 Runtime options	42
9.4 Data assimilation	48
9.5 Radiative Transfer/Forward Models	57
9.5.1 CRTM	58
9.6 Optimization	58
9.7 Parameter Estimation	59
9.7.1 Genetic Algorithm	59
9.7.2 Observations for Parameter Estimation	60
9.7.3 Walnut Gulch PBMR soil moisture data	60
9.8 Domain specification	61
9.8.1 Polar stereographic	61
9.8.2 Lambert conformal	61
9.8.3 Mercator	62
9.8.4 Gaussian	62
9.8.5 UTM	62
9.8.6 Cylindrical lat/lon	63
9.8.7 Gaussian	63
9.8.8 UTM	63
9.9 Parameters	64
9.10 Forcings	70
9.10.1 GDAS	70
9.10.2 GEOS	71
9.10.3 ECMWF	71
9.10.4 ECMWF Reanalysis	72
9.10.5 AFWA/AGRMET	72
9.10.6 PRINCETON	76
9.10.7 Rhone AGG	76
9.10.8 GSWP	77
9.10.9 GMAO GLDAS	78
9.10.10 GFS	78
9.10.11 GSWP1	79
9.11 Supplemental forcings	79
9.11.1 AGRMET radiation	79
9.11.2 AGRMET polar stereographic radiation	80
9.11.3 CMAP precipitation	80
9.11.4 CEOP station data	80
9.11.5 SCAN station data	81
9.11.6 NLDAS	81
9.11.7 SALDAS	83
9.11.8 TRMM 3B42RT precipitation	84
9.11.9 TRMM 3B42V6 precipitation	84
9.11.10 CMORPH precipitation	85
9.11.11 Stage II precipitation	85

9.11.12 Stage IV precipitation	85
9.11.13 D2PCPCAR	86
9.11.14 D2PCPOKL	86
9.11.15 NARR	86
9.11.16 GDAS3D	87
9.11.17 ARMS	87
9.12 Land surface models	87
9.12.1 Forcing only – Template	88
9.12.2 NCEP’s Noah	88
9.12.3 CLM 2.0	90
9.12.4 VIC	90
9.12.5 Mosaic	92
9.12.6 Hyssib	93
9.12.7 SiB2	94
9.12.8 Catchment	95
9.13 Model output configuration	96
10 Specification of Input Forcing Variables	98
11 Specification of Data Assimilation and Perturbation Attribute Files	100
11.1 Forcing perturbations specification	100
11.2 Model prognostic state perturbations specification	101
11.3 Observations perturbations specification	102
11.4 Bias options specifications	103
12 Specification of Parameter Estimation Attribute Files	104
12.1 Decision space specification for Genetic Algorithm	104
13 Model Output Specifications	105
A Cylindrical Lat/Lon Domain Example	109
B Polar Stereographic Domain Example	114
C Gaussian Domain Example	115
D Lambert Conformal Domain Example	119
E Mercator Domain Example	120
F UTM Domain Example	121
G configure.lis	122

H READ GRIB - Information and Instructions	123
H.1 SUBROUTINE GRIBGET (NUNIT, IERR)	123
H.2 SUBROUTINE GRIBREAD (NUNIT, DATA, NDATA, IERR)	124
H.3 SUBROUTINE GRIBHEADER (IERR)	124
H.4 SUBROUTINE GRIBDATA (DATARRAY, NDAT)	124
H.5 SUBROUTINE GRIBPRINT (ISEC)	125
H.6 SUBROUTINE GET_SEC1 (KSEC1)	125
H.7 SUBROUTINE GET_SEC2 (KSEC2)	125
H.8 SUBROUTINE GET_GRIDINFO (IGINFO, GINFO)	125
H.9 C-ROUTINE COPEN (UNIT, NUNIT, NAME, MODE, ERR, OFLAG)	125
H.10 SEC Header Array Information Tables	126
H.11 Additional information for setting up the READ_GRIB routines for use on Linux Machines	128
H.12 Example of Fortran code that calls READ_GRIB routines	129
I GRIB Output Information	132
I.1 SUBROUTINE GRIB1_SETUP	132
I.2 SUBROUTINE GRIB1_FINALIZE	133
I.3 SUBROUTINE DRV_WRITEVAR_GRIB	134
I.4 Output GRIB Variables	134
I.5 Example of Fortran code that calls GRIB output routines	136

1 Introduction

This is the LIS' User's Guide. This document describes how to download and install the code and data needed to run the LIS executable for LIS revision 6.0. It describes how to build and run the code, and finally this document also describes how to download output data-sets to use for validation.

This document consists of 10 sections, described as follows:

- 1 Introduction:** the section you are currently reading
- 2 Background:** general information about the LIS project
- 3 Preliminaries:** general information, steps, instructions, and definitions used throughout the rest of this document
- 4 Obtaining the Source Code:** the steps needed to download the source code
- 5 Building the Executable:** the steps needed to build the LIS executable
- 6 Running the Executable:** the steps needed to prepare and submit a run, also describes the various run-time configurations
- 7 Test Cases:** describes the LIS test cases.
- 8 Output Data Processing:** the steps needed to post-process generated output for visualization
- 9 LIS config File:** describes the user-configurable options.
- 13 Model Output List Table:** describes the user-configurable output variable options.

1.1 What's New

1.1.1 LIS 5.0-6.0

1. Modules have been restructured to streamline public and private interfaces
2. Restructured AGRMET processing—parallel support, lat/lon support.
3. This version now uses ESMF 3.1.0rp2.
4. Support for computational halos.
5. Allows mosaicing of different forcings concurrently (e.g. GDAS global + NLDAS over CONUS+SALDAS over south america, etc.)
6. Allows multiple overlays of different supplemental forcings (e.g. GDAS overlaid with NLDAS, AGRMET, STAGEIV)
7. Allows concurrent instances of data assimilation

8. Includes a highly configurable I/O interface (Allows unit conversions, temporal averaging, model-independent support for binary, Grib1 and NETCDF)
9. Includes support for 3d forcing (that includes the atmospheric profile) and a configurable specification of the forcing inputs
10. A dynamic bias estimation component (from NASA GMAO) has been added to the data assimilation subsystem.
11. Generic support for parameter estimation/optimization with the implementation of a heuristic approach using Genetic Algorithms.
12. New sources for data assimilation (using NASA and NESDIS retrievals of AMSRE soil moisture)
13. Support for real time GVF data from NESDIS and MODIS
14. A suite of upscaling algorithms to complement the existing spatial downscaling algorithms.
15. Support for new map projections - UTM
16. Support for forward modeling using radiative transfer models, and support for radiance based assimilation

1.1.2 LIS 4.2-5.0

1. This version includes the infrastructure for performing data assimilation using a number of different algorithms from simple approaches such as direct insertion to the more sophisticated ensemble kalman filtering.
2. More streamlined support for different architectures: A configuration based specification for the LIS makefile.
3. The data assimilation infrastructure utilizes the Earth System Modeling Framework (ESMF) structures. The LIS configuration utility has been replaced with the corresponding ESMF utility.

1.1.3 LIS 4.1-4.2

1. Completed implementation of AGRMET processing algorithms
2. Ability to run on polar stereographic, mercator, lambert conformal, and lat/lon projections
3. Updated spatial interpolation tools to support the transformations to/from the above grid projections
4. Switched to a highly interactive configurations management from the fortran namelist-based lis.crd style.

5. Streamlined error and diagnostic logging, in both sequential and parallel processing environments.
6. extended grib support; included the UCAR-based read-grib library
7. Support for new supplemental forcing analyses - Huffman, CMORPH

1.1.4 LIS 4.0.2 – 4.1

1. Preliminary AFWA support
2. Ability to run on a defined layout of processors.
3. Updates to plugins, preliminary implementation of alarms.
4. Definition of LIS specific environment variables.

1.1.5 LIS 4.0 – 4.0.2

1. GSWP-2 support – LIS can now run GSWP-2 experiments. Currently only CLM and Noah models have full support.
2. Updates to the 1km running mode.
3. Updates to the GDS running mode.

1.1.6 LIS 3.1 – 4.0

1. VIC 4.0.5 – LIS' implementation of VIC has been reinstated.

1.1.7 LIS 3.0 – 3.1

1. New domain-plugin support – facilitates creating new domains.
2. New domain definition support – facilitates defining running domains. Sub-domain selection now works for both MPI-based and non MPI-based runs.
3. New parameter-plugin support – facilitates adding new input parameter data-sets.
4. New LIS version of ipolates – facilitates creating new domains and base forcing data-sets.
5. Compile-time MPI support – MPI libraries are no longer required to compile LIS.
6. Compile-time netCDF support – netCDF libraries are no longer required to compile LIS.
7. New LIS time manager support – ESMF time manager was removed. ESMF libraries are not required in this version of LIS.

1.1.8 LIS 2.0 – 3.0

1. Running Modes – Now there is more than one way to run LIS. In addition to the standard MPI running mode, there are the GDS running mode and the 1 km running mode.
2. Sub-domain Selection – Now you are no longer limited to global simulations. You may choose any sub-set of the global domain to run over. See Section 9 for more details. (This is currently only available for the MPI-based running mode.)
3. Plug-ins – Now it is easy to add new LSM and forcing data-sets into the LIS driver. See LIS’ Developer’s Guide for more details.

2 Background

This section provides some general information about the LIS project.

2.1 LIS

Land Information System (LIS) is a flexible land surface modeling and data assimilation framework developed with the goal to integrate satellite- and ground-based observational data products and advanced land surface modeling techniques to produce optimal fields of land surface states and fluxes. The LIS infrastructure provides the modeling tools to integrate these observations with model forecasts to generate improved estimates of land surface conditions such as soil moisture, evaporation, snow pack, and runoff, at 1km and finer spatial resolutions and at one-hour and finer temporal resolutions. The fine scale spatial modeling capability of LIS allows it take advantage of the EOS-era observations, such as MODIS leaf area index, snow cover, and surface temperature, at their full native resolution. LIS features a high performance and flexible design, provides infrastructure for data integration and assimilation, and operates on an ensemble of land surface models (LSM) for extension over user-specified regional or global domains. LIS is designed using advanced software engineering principles to enable reuse and community sharing of modeling tools, data resources, and assimilation algorithms. The system is designed as an object-oriented framework, with abstractions defined for customization and extension to different applications. These extensible interfaces allow the incorporation of new domains, LSMs, land surface parameters, meteorological inputs, data assimilation and optimization algorithms. The extensible nature of these interfaces and the component style specification of the system allow rapid prototyping and development of new applications. These features enable LIS to serve both as a Problem Solving Environment (PSE) for hydrologic research to enable accurate global water and energy cycle predictions, and as a Decision Support System (DSS) to generate useful information for application areas including disaster management, water resources management, agricultural management, numerical weather prediction, air quality and military mobility assessment.

LIS currently includes a comprehensive suite of subsystems to support uncoupled and coupled land data assimilation. A schematic of the LIS framework with the associated subsystems are shown in the Figure below. The LIS-LSM subsystem, which is the core of LIS, supports high performance, interoperable and portable land surface modeling with a suite of community land surface models and input data. Further, the LIS-LSM subsystem is designed to encapsulate the land surface component of an Earth System model. The LIS-WRF subsystem supports coupled land-atmosphere modeling through both one-way and two-way coupling to the WRF atmospheric model, leading to a hydrometeorological modeling capability that can be used to evaluate the impact of land surface processes on hydrologic prediction. The Data Assimilation (LIS-DA) subsystem supports multiple data assimilation algorithms that are focused on generating improved estimates of hydrologic model states. Finally, the Opti-

mization (LIS-OPT) subsystem (which is under development) supports a suite of advanced optimization and uncertainty modeling tools in LIS.

2.2 LIS core

The central part of LIS software system is the LIS core that controls program execution. The LIS core is a model control and input/output system (consisting of a number of subroutines, modules written in Fortran 90 source code) that drives multiple offline one-dimensional LSMs. The one-dimensional LSMs such as CLM and Noah, apply the governing equations of the physical processes of the soil-vegetation-snowpack medium. These land surface models aim to characterize the transfer of mass, energy, and momentum between a vegetated surface and the atmosphere. When there are multiple vegetation types inside a grid box, the grid box is further divided into “tiles”, with each tile representing a specific vegetation type within the grid box, in order to simulate sub-grid scale variability.

The execution of the LIS core starts with reading in the user specifications, including the modeling domain, spatial resolution, duration of the run, etc. Section 6 describes the exhaustive list of parameters specified by the user. This is followed by the reading and computing of model parameters. The time loop begins and forcing data is read, time/space interpolation is computed and modified as necessary. Forcing data is used to specify the boundary conditions to the land surface model. The LIS core applies time/space interpolation to convert the forcing data to the appropriate resolution required by the model. The selected model is run for a vector of “tiles” and output and restart files are written at the specified output interval.

Some of the salient features provided by the LIS core include:

- Vegetation type-based “tile” or “patch” approach to simulate sub-grid scale variability.
- Makes use of various satellite and ground-based observational systems.
- Derives model parameters from existing topography, vegetation, and soil coverages.
- Extensible interfaces to facilitate incorporation of new land surface models, forcing schemes.
- Uses a modular, object oriented style design that allows “plug and play” of different features by allowing user to select only the components of interest while building the executable.
- Ability to perform regional modeling (only on the domain of interest).
- Provides a number of scalable parallel processing modes of operation.

Please refer to the software design document for a detailed description of the design of LIS core. The LIS reference manual provides a description of the

extensible interfaces in LIS. The “plug and play” feature of different components is described in this document.

DRAFT

3 Preliminaries

This section provides some preliminary information to make reading this guide easier.

Commands are written like this:

```
% cd /path/to/LISv6.0  
% ls  
“... compiler flags, then run gmake.”
```

File names are written like this:

/path/to/LISv6.0/src

You need to create a working directory on your system to install and run LIS in. Let's call this directory */path/to/LISv6.0/*. Throughout the rest of this document, this directory shall be referred to as *\$WORKING*. You should create a directory to run LIS in. This directory can be created anywhere on your system, but, in this document, we shall refer to this running directory as *\$WORKING/run*.

4 Obtaining the Source Code

This section describes how to obtain the source code needed to build the LIS executable.

The source code is maintained in a Subversion repository; thus, you need the Subversion client (svn) to obtain this code. If you need any help regarding Subversion, please go to <http://subversion.tigris.org/>.

4.1 Downloading the Source Code

To obtain the source code needed for LIS revision 6.0 you must first obtain access to the LIS source code repository. To obtain access you must contact the LIS team. Once you have access to the repository, you will be given the correct Subversion command to run to checkout the source code.

Please note that LIS' web-site is under reconstruction.

1. Goto the `$WORKING` directory.
`% cd $WORKING`
2. Check out the source code into a directory called `src`.

For the public version, run the following command:

```
% svn checkout https://flood.gsfc.nasa.gov/svn/6/public src
```

For the development version, run the following command:

```
% svn checkout https://flood.gsfc.nasa.gov/svn/6/development src
```

4.2 Source files

Checking out the LIS source code (according the instructions in Section 4) will create a directory named `src`. The structure of `src` is as follows:

- *arch*
Directory containing the configurable options for building the LIS executable
- *baseforcing*
Top level directory for base meteorological forcing methods, which includes the following implementations
 - *afwa*
AFWA AGRMET meteorlogical forcing data
 - *ecmwf*
ECMWF meteorlogical forcing data
 - *ecmwfreanal*
ECMWF reanalysis meteorlogical forcing data based on Berg et al. (2003).

- *gdas*
NCEP GDAS meteorlogical forcing data
 - *geos*
NASA GEOS meteorlogical forcing data
 - *gfs*
NCEP GFS meteorlogical forcing data
 - *gswp*
Global Soil Wetness Project-2 meteorlogical forcing data
 - *gswp1*
Global Soil Wetness Project-1 meteorlogical forcing data
 - *princeton*
Renalalysis product from Princeton University (Sheffield et al. (2006))
 - *rhone*
Rhone-AGG meteorlogical forcing data
 - *template*
An empty template for meteorlogical forcing data implementations
- *configs*
some sample LIS configuration files
- *core*
core routines in LIS
- *dataassim*
Top level directory for data assimilation support, which includes the following subcomponents
 - *algorithm*
Directory containing the following data assimilation algorithm implementations:
 - * *di*: direct insertion algorithm for data assimilation
 - * *gmaoenkf*: NASA GMAO's Ensemble Kalman Filter algorithm for data assimilation
 - *biasEstimation*
Directory containing the following dynamic bias estimation algorithms:
 - * *gmaoBE*: NASA GMAO's dynamic bias estimation algorithm
 - *obs*
Directory containing the following observation handlers for data assimilation:
 - * *AMREsm hdf*: AMSRE Level-3 daily gridded soil moisture product in HDF4/HDFEOS format
 - * *ANSA*: AFWA NASA snow product in HDF5 format

- * *ISCCP_Tskin*: ISCCP skin temperature product in binary format
- * *MODISsca*: MODIS snow cover area product in HDF4/HDFEOS format
- * *NASA_AMSREsm*: NASA AMSRE soil moisture data in binary format
- * *NESDIS_AMSREsm*: NESDIS AMSRE soil moisture data in binary format
- * *SNODEP*: AFWA snowdepth data in Grib1 format

This directory also includes the following synthetic data handler examples:

- * *syntheticsm1*: synthetic soil moisture data handler (produced from a Noah LSM run)
- * *syntheticsm2*: synthetic soil moisture data handler (produced from a Catchment LSM run)
- * *syntheticlst1*: synthetic land surface temperature data handler
- * *syntheticswe1*: synthetic snow water equivalent data handler
- * *multisynsmobs*: synthetic soil moisture data with multiple observation types
- *perturb*
Directory containing the following perturbation algorithm implementations
 - * *static*: Static gaussian normal perturbation algorithm
 - * *gmaopert*: NASA GMAO's perturbation algorithm
- *domains* Directory containing the domains in the following map projections / custom grids
 - *afwa*: AGRMET domain (1/2 degree or 1/4 degree equidistant cylindrical grid)
 - *catchment*: ??
 - *gaussian*: NCEP gaussian grids
 - *gswp*: GSWP equidistant cylindrical grids
 - *hrap*: Hydrologic Rainfall Analysis Project polar stereographic grid
 - *lambert*: Lambert conformal grids
 - *latlon*: Equidistant cylindrical grids
 - *merc*: Mercator grids
 - *polar*: Polar stereographic grids
 - *UTM*: Universal Transverse Mercator grids
- *interp*
Generic spatial and temporal interpolation routines

- *lib*
External libraries supplied with the LIS source code
- *lsms*
Directory containing implementations of the following land surface models
 - *Catchment*: NASA GMAO Catchment land surface model
 - *clm2*: NCAR community land model version 2.0
 - *clm3.5*: NCAR community land model version 3.5
 - *hyssib*: NASA HySSIB land surface model
 - *mlbc*: ??
 - *mosaic*: NASA Mosaic land surface model
 - *noah.2.7.1*: NCEP Noah land surface model version 2.7.1
 - *noah.3.0*: NCEP Noah land surface model version 3.0
 - *noah.3.1*: NCEP Noah land surface model version 3.1
 - *place*: PLACE land surface model
 - *sacramento*: OHD Sacramento model
 - *snow17*: OHD snow17 model
 - *sacsnow17*: Combined use of Sacramento and snow17 models
 - *SiB2*: SiB2 land surface model
 - *tessel*: ECMWF Tessel land surface model
 - *vic*: Variable Infiltration Capacity model from Princeton university and University of Washington
 - *template*: An empty template for land surface model implementations

Each of these LSM directories contain specific plugin interfaces related to (1) coupling to WRF and GCE models, (2) Data assimilation instances and (3) Parameter estimation instances. These routines defined for Noah land surface model version 2.7.1 are shown below. Note that similar routines are implemented in other LSMs.

1. Coupling interfaces:
 - *cpl_gce*: Routines for coupling Noah with GCE
 - *cpl_wrf*: Routines for coupling Noah with WRF
2. Data assimilation interfaces:
 - *da_modissa*: Noah routines related to the assimilation of MODIS snow cover area
 - *da_snodep*: Noah routines related to the assimilation of AFWA SNOSEP observations
 - *da_snow*: Noah routines related to the assimilation of snow water equivalent observations

- *da_tskin*: Noah routines related to the assimilation of land surface temperature observations
 - *da_soilm*: Noah routines related to the assimilation of soil moisture observations
 - *da_multism*: Noah routines related to the assimilation of data with multiple soil moisture observation types
- 3. Parameter estimation interfaces:
 - *pe_soils*: Noah routines related to the estimation of soil properties through parameter estimation
 - *pe_z0*: Noah routines related to the estimation of roughness length through parameter estimation
- *make*
 Makefile and needed header files for building LIS executable
- *offline*
 Contains the main program for the offline mode of operation
- *optimization*
 Top level directory for optimization support, which includes the following subcomponents
 - *algorithm*
 Directory containing the following optimization algorithm implementations
 - * *LM*: Levenberg - Marquardt gradient search algorithm
 - * *GA*: Single objective Genetic Algorithm
 - * *SCE-UA*: Shuffled Complex Evolutionary Algorithm
- *paramestim*
 Top level directory for parameter estimation support, which includes the following subcomponents
 - *objfunc*
 Directory containing the following objective function evaluation methods
 - * *LS*: Least squares based objective function
 - *obs*
 Directory containing the following observation handlers for parameter estimation
 - * *ISCCP_Tskin*: ISCCP land surface temperature observations
 - * *wgPBMRsm*: PBMR soil moisture data for the Walnut Gulch watershed
- *params*
 Directory containing implementations of the following land surface model parameters

- *albedo*: Routines for handling albedo data products
 - *gfrac*: Routines for handling green vegetation fraction data products
 - *lai*: Routines for handling Leaf/Stem area index data products
 - *landcover*: Routines for handling landcover and landmask data products
 - *soils*: Routines for handling soil property data products
 - *tbot*: Routines for handling bottom temperature data products
 - *topo*: Routines for handling topography data products
- *plugins*
Modules defining the function table registry of extensible functionalities
- *rtrms*
Directory containing coupling routines to the following radiative transfer models
 - *CRTM*
Routines to handle coupling to the JCSDA Community Radiative Transfer Model
- *runmodes*
Directory containing the following running modes in LIS
 - *agrmetmode*: Routines to manage the program flow in the AFWA operational mode
 - *gce_cpl_mode*: Routines to manage the program flow in the coupled LIS-GCE mode
 - *wrf_cpl_mode*: Routines to manage the program flow in the coupled LIS-WRF mode
 - *paramEstimation*: Routines to manage the program flow in the parameter estimation mode
 - *retrospective*: Routines to manage the program flow in the retrospective analysis mode
 - *rtmmode*: Routines to manage the program flow when a forward model integration using a radiative transfer model is employed
- *suppforcing*
Directory containing the following supplemental forcing implementations
 - *3B42RT*: Routines for handling the TRMM 3B42RT precipitation product
 - *3B42V6*: Routines for handling the TRMM 3B42V6 precipitation product
 - *agrrad*: Routines for handling the AGRMET radiation product

- *agrradps*:
 - *arms*: Routines for handling the Walnut Gulch meteorological station data
 - *ceop*: Routines for handling the CEOP meteorological station data
 - *cmap*: Routines for handling the CMAP precipitation product
 - *cmorph*: Routines for handling the CMORPH precipitation product
 - *gdas3d*: Routines for handling the GDAS 3d (including the atmospheric profile) data
 - *narr*: Routines for handling the North American Regional Reanalysis (3d) data
 - *nldas*: Routines for handling the North American Land Data Assimilation System forcing product
 - *nldas2*: Routines for handling the NLDAS2 forcing product
 - *saldas*: Routines for handling the South American Land Data Assimilation System forcing product
 - *scan*: Routines for handling the Soil Climate Analysis Network precipitation product
 - *stg2*: Routines for handling the NCEP Stage IV QPE precipitation product
 - *stg4*: Routines for handling the NCEP Stage II precipitation product
- *testcases*
testcases for verifying various functionalities
 - *utils*
Miscellaneous helpful utilities

Source code documentation may be found on LIS' web-site on LIS' "Public Release Home Page". Follow the "LIS 6.0 Source Code Documentation" link.

5 Building the Executable

This section describes how to build the source code and create LIS' executable – named LIS.

5.0.1 Development Tools

This code has been compiled and run on Linux PC (Intel/AMD based) systems, IBM AIX systems, and SGI Altix systems. These instructions expect that you are using such a system. In particular you need

- Linux
 - Absoft's Pro Fortran Software Development Kit, version 10.0
or
Lahey/Fujitsu's Fortran 95 Compiler, release L6.00c
 - GNU's C and C++ compilers, gcc and g++, version 3.3.3
 - GNU's make, gmake, version 3.77
- IBM
 - XL Fortran version 10.1.0.6
 - GNU's make, gmake, version 3.77
- SGI Altix
 - Intel Fortran Compiler version 10.1.017
 - GNU's make, gmake, version 3.77

5.0.2 Required Software Libraries

In order to build the LIS executable, the following libraries must be installed on your system:

- Earth System Modeling Framework (ESMF) version 3.1.0rp2 (<http://www.esmf.ucar.edu/>)
Please read the ESMF User's Guide for details on compiling ESMF with MPI support and without MPI support ("mpuni").

5.0.3 Optional Software Libraries

The following libraries are not required to compile LIS. They are used to extend the functionality of LIS.

- Message Passing Interface (MPI) – If you wish to run the MPI-based running mode
 - vendor supplied, or
 - MPICH version 1.2.7p1 (<http://www-unix.mcs.anl.gov/mpi/mpich1/>)

- If you choose to have NETCDF support, please download the netcdf source (<http://www.unidata.ucar.edu/software/netcdf/>) and compile the source to generate the NETCDF library.
- If you choose to have HDF (version 4 or 5) support, please download the hdf source (<http://hdf.ncsa.uiuc.edu/>)

To install these libraries, follow the instructions provided at the various URL listed above.

Note: Due to the mix of programming languages (Fortran and C) used by LIS, you may run into linking errors when building the LIS executable.

When compiling code using Absoft's Pro Fortran SDK, set the following compiler options:

`-YEXT_NAMES=LCS -s -YEXT_SFX=_ -YCFRL=1`

These must be set for each of the above libraries.

5.0.4 Build Instructions

1. Perform the steps described in Section 4 to obtain the source code.
2. Goto the `$WORKING/src/arch` directory. A number of files named `configure.lis.*` exist in this directory. Each file contains the configurable options that are specific for each architecture and compiler. For example, the file `configure.lis.aix` contains the set of configurable options for an IBM AIX platform. Depending on your choice of platform, edit this file or create a new file for your platform with the set of options. The following is a list of variables that need to be specified in the `configure.lis` file.

Variable	Description
FC	fortran 90 compiler
FC77	fortan 77 compiler
LD	fortran linker
CC	C compiler
AR	program to create a library archive
INC_NETCDF	path to NETCDF header files
LIB_NETCDF	path to NETCDF library files
INC_HDF	path to HDF header files
LIB_HDF	path to HDF library files
INC_HDFEOS	path to HDFEOS header files
LIB_HDFEOS	path to HDFEOS library files
INC_CRTM	path to CRTM header files
LIB_CRTM	path to CRTM library files
LIB_MPI	path to mpi libraries
INC_MPI	path to mpi header files
LIB_ESMF	path to esmf libraries
MOD_ESMF	path to esmf modules
CFLAGS	flags for C compiler
FFLAGS	flags for Fortran 90 compiler
FFLAGS77	flags for Fortran 77 compiler
LDFLAGS	flags for linker

If the user choses to compile and run on a single processor with no MPI, the options in the *configure.lis* file should be specified accordingly. Specifying the compiler preprocessor flag *-DSPMD* enables the compiling of the code with MPI support. Removing this flag produces a serial version of LIS.

3. Compile the new GRIB library, *libw3.a*. You must edit the *Makefile* located in *\$WORKING/src/lib/w3lib*. Uncomment the appropriate block of compiler flags, then run **gmake**.
4. Compile the new GRIB library, *griblib.a*. You must edit the *Makefile* located in *\$WORKING/src/lib/grib*. Uncomment the appropriate block of compiler flags, then run **gmake**.
5. Compile the new GRIB library, *read_grib*. You must edit the *Makefile* located in *\$WORKING/src/lib/read_grib*. Uncomment the appropriate block of compiler flags, then run **gmake <arch>**, where *<arch>* is the appropriate architecture or compiler label. Running **gmake** will produce a list of acceptable values. For example, to compile on a system using the Intel Fortran compiler, run **gmake ifc**. Please refer to the Appendix H for helpful suggestions and instructions. If you are on an IBM AIX system, use the *read_grib.aix* library.
6. All the included libraries are generated. Copy the appropriate *configure.lis.** file to *\$WORKING/src/make/configure.lis* and edit this *configure.lis* file to make sure the file paths are specified correctly.

7. Compile the dependency generator, *makdep*. Change directory into *\$WORKING/src/make/MAKDEP*. Run **gmake**.
8. Compile the LIS source code.
 - (a) Change directory into *\$WORKING/src/make*.


```
% cd $WORKING/src/make
```
 - (b) Edit the *misc.h* file to specify if NETCDF support should be included.
 If **define USE_NETCDF** is set, NETCDF support will be included.
 To disable NETCDF support, edit the *misc.h* file to specify **UNDEF USE_NETCDF**.
 - (c) Edit the *misc.h* file to specify if HDF4 support should be included.
 If **define USE_HDF4** is set, HDF4 support will be included.
 To disable HDF4 support, edit the *misc.h* file to specify **UNDEF USE_HDF4**.
 - (d) Edit the *misc.h* file to specify if HDF5 support should be included.
 If **define USE_HDF5** is set, HDF5 support will be included.
 To disable HDF5 support, edit the *misc.h* file to specify **UNDEF USE_HDF5**.
 - (e) Run the make command.


```
% gmake
```
 - (f) Finally, copy the *LIS* executable into your running directory, *\$WORKING/run*.

See Appendix G to see a *configure.lis* file.

5.1 Generating documentation

LIS code uses the ProTex documenting system Sawyer and da Silva (1997). The documentation in LATEX format can be produced by using the *doc.csh* in the *\$WORKING/src/utils* directory. This command produces documentation, generating a number of LATEX files. These files can be easily converted to pdf or html formats using utilites such as *pdflatex* or *latex2html*.

6 Running the Executable

This section describes how to run the LIS executable.

The single-process version of LIS is executed by the following command issued in the `$WORKING/run/` directory.

```
% ./LIS
```

The parallel version of LIS must be run through an `mpirun` script or similar mechanism. Assuming that MPI is installed correctly, the LIS simulation is carried out by the following command issued from in the `$WORKING/run/` directory.

```
% mpirun -np N ./LIS
```

The `-np N` flag indicates the number of processes to use in the run, where you replace `N` with the number of processes to use. On a multiprocessor machine, the parallel processing capabilities of LIS can be exploited using this flag.

To customize your run, you must modify the `lis.config` configuration file. See Section 9 for more information.

7 Test Cases

This section describes how to obtain and how to use the test cases provided by the LIS team.

The test cases are comprised of three parts: a *testcases* sub-directory include in the LIS source code, input data, and output data.

7.1 The *testcases* Sub-directory

The layout of the *testcases* sub-directory matches the layout of the top-level *src* directory. For example, LIS contains support for processing GDAS forcing data. These routines are in *src/baseforcing/gdas*. The test case for GDAS is in *src/testcases/baseforcing/gdas*.

These test case sub-directories contain several files to help you test LIS. For example, the *src/testcases/baseforcing/gdas* test case contains three files: *README*, *lis.config*, *MODEL_OUTPUT_LIST.TBL*, and *output.ctl*.

The file, *README*, contains instructions on how to run the test case. The file, *lis.config*, is a configuration file to set the test case. The file, *MODEL_OUTPUT_LIST.TBL*, is a configuration file to set the output for the test case. The file, *output.ctl*, is a GrADS descriptor file. This file is used with GrADS to plot the output data. You may also read this file to obtain metadata regarding the structure of the output files. This metadata is useful in helping you plot the output using a different program.

7.2 Test Cases Input

For each test case, the LIS team has created a corresponding input data file that contains all the required data for running the test case.

To obtain the input data for a test case:

1. Go to LIS’ “Public Release Home Page”
Go to <http://lis.gsfc.nasa.gov/>
Follow the “Source Codes” link.
Follow the “LIS 6.0 Code Release” link.
2. From LIS’ “Public Release Home Page”
Follow the “LIS testcases page” link found in the “Input Data” section.

7.3 Test Cases Output

For each test case, the LIS team has created a corresponding output data file that contains all the output data produced from the test case.

To obtain the output data for a test case:

1. Go to LIS' "Public Release Home Page"
Go to <http://lis.gsfc.nasa.gov/>
Follow the "Source Codes" link.
Follow the "LIS 6.0 BETA Code Release" link.
2. From LIS' "Public Release Home Page"
Follow the "LIS testcases page" link found in the "Output Data" section.

7.3.1 Output Example

For example, output data for the "Noah LSM TEST" contains:

```
OUTPUT/EXP111/NOAHstats.d01.stats  
OUTPUT/EXP111/NOAH/2002/20021029/200210290300.d01.gs4r  
OUTPUT/EXP111/NOAH/2002/20021029/200210290600.d01.gs4r  
OUTPUT/EXP111/NOAH/2002/20021029/200210290900.d01.gs4r  
OUTPUT/EXP111/NOAH/2002/20021029/200210291200.d01.gs4r  
OUTPUT/EXP111/NOAH/2002/20021029/200210291500.d01.gs4r  
OUTPUT/EXP111/NOAH/2002/20021029/200210291800.d01.gs4r  
OUTPUT/EXP111/NOAH/2002/20021029/200210292100.d01.gs4r  
OUTPUT/EXP111/NOAH/2002/20021030/200210300000.d01.gs4r  
OUTPUT/EXP111/NOAH/2002/20021030/LIS.E111.200210300000.d01.Noahrst  
OUTPUT/EXP111/NOAH/2002/20021030/200210300300.d01.gs4r  
OUTPUT/EXP111/NOAH/2002/20021030/200210300600.d01.gs4r  
OUTPUT/EXP111/NOAH/2002/20021030/200210300900.d01.gs4r  
OUTPUT/EXP111/NOAH/2002/20021030/200210301200.d01.gs4r  
OUTPUT/EXP111/NOAH/2002/20021030/200210301500.d01.gs4r  
OUTPUT/EXP111/NOAH/2002/20021030/200210301800.d01.gs4r  
OUTPUT/EXP111/NOAH/2002/20021030/200210302100.d01.gs4r  
OUTPUT/EXP111/NOAH/2002/20021031/200210310000.d01.gs4r  
OUTPUT/EXP111/NOAH/2002/20021031/LIS.E111.200210310000.d01.Noahrst  
OUTPUT/EXP111/NOAH/2002/20021031/LIS.E111.200210310100.d01.Noahrst
```

The file, *OUTPUT/EXP111/NOAHstats.d01.stats*, contains statistics from the run.

The files labelled like *OUTPUT/EXP111/NOAH/2002/20021029/200210290300.d01.gs4r* contain the output from the run. Read the *output.ctl* file contained in the appropriate *testcases* sub-directory of the LIS source code for metadata pertaining to these output files.

The files labelled like

OUTPUT/EXP111/NOAH/2002/20021031/LIS.E111.200210310000.d01.Noahrst are restart files. They may be used to continue or restart a run. The data are valid for the date and time indicated by the date-stamp in the file name. For example, the restart data in this file, *OUTPUT/EXP111/NOAH/2002/20021031/LIS.E111.200210310000.d01.Noahrst*, are valid for 2002-10-31T00:00:00.

These output data files are large and require post-processing before reading them, see Section 8.

8 Output Data Processing

This section describes how to process the generated output in various formats. The generated output can be written in a Fortran binary, GRIB, or NetCDF format. See Section 9.3 for more details.

8.1 Binary output format

The output data-sets created by running the LIS executable are written into sub-directories of the `$WORKING/run/OUTPUT/` directory (created at run-time). These sub-directories are named such as `EXP999`. See Section 9.3.

The output data consists of ASCII text files and model output in a binary format. For the Fortran binary format, LIS writes the output data as 4-byte REALs in sequential access mode.

For example, assume that you performed the Noah test case.

This run will produce a `$WORKING/run/OUTPUT/EXP111/` directory. This directory will contain:

File Name	Synopsis
Noahstats.dat	Statistical summary of output
NOAH	Directory containing output data

The `NOAH` directory will contain sub-directories of the form `YYYY/YYYYMMDD`, where `YYYY` is a 4-digit year and `YYYYMMDD` is a date written as a 4-digit year, 2-digit month and a 2-digit day; both corresponding to the running dates of the simulation.

For this example, `NOAH` will contain a `2002/20021030` sub-directory.

Its contents are the output files generated by the executable. They are:

```
200210300000.d01.gs4r
200210300300.d01.gs4r
200210300600.d01.gs4r
200210300900.d01.gs4r
200210301200.d01.gs4r
200210301500.d01.gs4r
200210301800.d01.gs4r
200210302100.d01.gs4r
```

Note, each file-name contains a date-stamp marking the year, month, day, hour, and minute that the data corresponds to. The output data files for other land surface models are similar.

The actual contents of the output files depend on the settings in the `lis.config` configuration file and the “Model output attributes file” file defined within the

lis.config configuration file. See Section 9.13. The order in which the variables are written is the same order as in the statistical summary file; e.g., *Noah-stats.dat*.

The generated output can be written in a 2-D grid format or as a 1-d vector. See Section 9.3 for more details. If written as a 1-d vector, the output must be converted into a 2-d grid before it can be visualized. This is left as an exercise for the reader.

8.2 GRIB1 output format

GRIB1 is a self-describing data format. The output files produced in GRIB1 can be inspected by using the utility *wgrib*.

8.3 NetCDF output format

NetCDF is a self-describing format. The output files produced in NetCDF can be inspected by using the utility *ncdump*.

9 LIS config File

This section describes the options in the *lis.config* file.

9.1 Overall driver options

Running mode: specifies the running mode used in LIS. Acceptable values are:

Value	Description
1	retrospective mode
2	AFWA AGRMET mode
3	Coupled WRF mode
4	Coupled GCE mode
5	Coupled GFS mode
6	parameter estimation mode
7	RTM forward mode

Running mode: 1

Domain type: specifies the LIS domain used for the run Acceptable values are:

Value	Description
1	Lat/Lon projection with SW to NE data ordering
2	Mercator projection with SW to NE data ordering
3	Lambert conformal projection with SW to NE data ordering
4	Gaussian domain
5	Polar stereographic projection with SW to NE data ordering
6	AFWA lat/lon 0.5 degree/0.25 degree domain with no subgrid tiling
7	UTM domain
8	HRAP domain
10	Catchment based domain
11	GSWP domain

Domain type: 1

Number of nests: specifies the number of nests used for the run. Values 1 or higher are acceptable. The maximum number of nests is limited by the amount of available memory on the system. The specifications for different nests are done using white spaces as the delimiter. Please see below for further explanations. Note that all nested domains should run on the same projection and same land surface model.

Number of nests: 1

Land surface model: specifies the land surface model to run. Acceptable values are:

Value	Description
0	template lsm
1	Noah 2.7.1
2	CLM 2.0
3	VIC
4	mosaic
5	hyssib
6	sib2
7	catchment
8	sacramento
9	snow17
10	sacramento+snow17
11	sib3
12	mlbc
13	csu
14	place
21	noah 3.1

Land surface model: 1

Base forcing source: specifies the forcing data source for the run. Acceptable values are:

Value	Description
0	template base forcing
1	GDAS
2	GEOS
3	ECMWF
5	GSPW2
6	ECMWF Reanalysis
7	AGRMET
9	Princeton
11	Rhone-AGG forcing
12	GSPW1
13	GMAO GLDAS
14	GFS

Base forcing source: 1

Number of base forcing variables: specifies the number of variables in the base forcing

Number of base forcing variables: 10

Use elevation correction (base forcing): specifies whether to use elevation correction for base forcing.

Acceptable values are:

Value	Description
0	Do not use elevation correction for forcing
1	Use lapse rate correction for forcing
2	Use micro met correction for forcing – not supported

Use elevation correction (base forcing): 1

Spatial interpolation method (base forcing): specifies the type of interpolation scheme to apply to the base forcing data. Acceptable values are:

Value	Description
1	bilinear scheme
2	conservative scheme
3	neighbour scheme

Bilinear interpolation uses 4 neighboring points to compute the interpolation weights. The conservative approach uses 25 neighboring points. If the conservative option is turned on, it is used to interpolate the precip field only (to conserve water). Other fields will still be interpolated with the bilinear option.

Spatial interpolation method (base forcing):

1

Temporal interpolation method (base forcing): specifies the type of temporal interpolation scheme to apply to the base forcing data. Acceptable values are:

Value	Description
1	linear scheme
2	ueber next scheme

The linear temporal interpolation method computes the temporal weights based on two points. Ubernext computes weights based on three points. Currently the ubernext option is implemented only for the GSWP forcing.

Temporal interpolation method (base forcing):

1

Number of supplemental forcing sources: specifies the number of supplemental forcing datasets to be used. Acceptable values are 0 or higher.

Number of supplemental forcing sources: 0

Supplemental forcing sources: specifies the supplemental forcing data sources for the run. The values should be specified in a column format, in the order in which they should be overlaid.

Acceptable values for the sources are:

Value	Description
0	None
1	AGRMET Radiation
2	CMAP
3	AGRMET radiation in PS projection
4	NLDAS
5	NLDAS2
6	CEOP
7	SCAN
10	SALDAS
11	TRMM 3B42RT
12	TRMM 3B42V6
14	CMORPH
15	Stage II
16	Stage IV
17	D2PCPCAR
18	D2PCPOKL
19	GDAS3D
20	NARR
21	ARMS

Supplemental forcing sources: 0

Number of supplemental forcing variables: specifies the number of forcing variables in the supplemental forcing. This should also be specified in a space delimited column format that corresponds to the order of the supplemental forcing sources.

Number of supplemental forcing variables: 0

Use elevation correction (supplemental forcing): specifies whether to use elevation correction for the supplemental forcing sources. The specification should be done in a space delimited column format that corresponds to the order of the supplemental forcing sources.

Acceptable values are:

Value	Description
0	Do not use elevation correction for forcing
1	Use lapse rate correction for forcing
2	Use micro met correction for forcing – not supported

Use elevation correction (supplemental forcing): 1

Spatial interpolation method (supplemental forcing): specifies the type of interpolation scheme to apply to the supplemental forcing data. The specification should be made in a space delimited column format that corresponds to the order of the supplemental forcing sources. Acceptable values are:

Value	Description
1	bilinear scheme
2	conservative scheme
3	neighbour scheme

Bilinear interpolation uses 4 neighboring points to compute the interpolation weights. The conservative approach uses 25 neighboring points. If the conservative option is turned on, it is used to interpolate the precip field only (to conserve water). Other fields will still be interpolated with the bilinear option.

Spatial interpolation method (supplemental forcing):

1

Temporal interpolation method (supplemental forcing): specifies the type of temporal interpolation scheme to apply to the supplemental forcing sources. The specification should be made in a space delimited column format that corresponds to the order of the supplemental forcing sources. Acceptable values are:

Value	Description
1	linear scheme
2	uber next scheme

The linear temporal interpolation method computes the temporal weights based on two points. Ubernext computes weights based on three points. Currently the ubernext option is implemented only for the GSWP forcing.

Temporal interpolation method (supplemental forcing):

1

9.2 Parameter options

The following options list the choice of parameter maps to be used

Map projection of parameter data: specifies the map projection of the parameter datasets. Note that the grid description options for the parameters will be different for different map projections

Acceptable values are:

Value	Description
0	Equidistant cylindrical (lat/lon)
4	Gaussian
5	Polar Stereographic
6	UTM projection

Map projection of parameter data: 0

Landcover data source: specifies the usage of soils data in the run. Acceptable values are:

Value	Description
1	use the UMD landcover
2	use the USGS landcover data
3	use the GFS landcover data
4	use the IGBP landcover data

Landcover data source: 1

Use soil texture: specifies the usage of soil texture data in the run. Acceptable values are:

Value	Description
0	use a sand/silt/clay percentage map
1	use a texture map

Use soil texture: 0

Soil data source: specifies the source of soil parameters in the run. Acceptable values are:

Value	Description
1	use FAO based maps
2	use STATSGO based maps
3	use GFS based maps

Soil data source: 1

Soil color data source: specifies the source of soil color data in the run.
Acceptable values are:

Value	Description
0	do not use soil color
1	use FAO based map
1	use STATSGO based map

Soil color data source: 0

Elevation data source: specifies topography data source for the run. Acceptable values are:

Value	Description
0	do not use
1	GTOPO30 based
2	Reserved
3	GFS based

Elevation data source: 1

slope map: specifies the slope of the LIS grid. Acceptable values are:

Value	Description
0	do not use
1	GTOPO30 based

Slope data source: 0

aspect map: specifies the aspect of the LIS grid. Acceptable values are:

Value	Description
0	do not use
1	GTOPO30 based

Aspect data source: 0

curvature map: specifies the curvature of the LIS grid. Acceptable values are:

Value	Description
0	do not use
1	GTOPO30 based

Curvature data source: 0

LAI data source: specifies the LAI data source for the run. Acceptable values are:

Value	Description
0	do not use
1	AVHRR-based LAI
2	MODIS-based LAI
3	Reserved
4	Catchment-based LAI
5	Tiled AVHRR-based LAI

LAI data source: 0

SAI data source: specifies the SAI data source for the run. Acceptable values are:

Value	Description
0	do not use
1	AVHRR-based SAI
2	MODIS-based SAI
3	Reserved
4	Catchment-based SAI
5	Tiled AVHRR-based SAI

SAI data source: 0

Albedo data source: specifies if albedo data is to be used in the run. Acceptable values are:

Value	Description
0	Do not read albedo data
1	NCEP Climatology
2	Reserved
3	GFS

Albedo data source: 1

Greenness data source: specifies if greenness fraction data is to be used in the run. Acceptable values are:

Value	Description
0	Do not read gfrac data
1	NCEP Climatology
2	Catchment based data
3	GFS

Greenness data source: 1

Porosity data source: specifies if soil porosity data is to be used in the run. Acceptable values are:

Value	Description
0	Do not read porosity data
1	FAO-based data

Porosity data source: 0

Ksat data source: specifies if hydraulic conductivity data is to be used in the run. Acceptable values are:

Value	Description
0	Do not read ksat data
1	FAO-based data

Ksat data source: 0

B parameter data source: specifies if the b parameter data is to be used in the run. Acceptable values are:

Value	Description
0	Do not read b parameter data
1	FAO-based data

B parameter data source: 0

Quartz data source: specifies if the quartz data is to be used in the run.
Acceptable values are:

Value	Description
0	Do not read quartz data

Quartz data source: 0

9.3 Runtime options

Experiment code: specifies the “experiment code number” for the run. It is used in constructing the name of the output directory for the run. Acceptable values are any name using up to 3 characters.

Experiment code: '111'

Number of veg types: specifies the number of vegetation types used in the landcover data. Acceptable values are:

Value	Description
13	UMD-based landcover types
16	IGBP-based landcover types
30	USGS-based landcover types

Number of veg types: 13

Forcing variables list file: specifies the file containing the list of forcing variables to be used. Please refer to the sample forcing_variables.txt (Section 10) file for a complete specification description.

Forcing variables list file: ./input/forcing_variables.txt

Output forcing: specifies whether to output the processed forcing variables.
Acceptable values are:

Value	Description
0	Do not output forcing variables
1	Do output forcing variables

Output forcing: 1

Output parameters: specifies whether to output the processed parameter variables.
Acceptable values are:

Value	Description
0	Do not output parameter variables
1	Do output parameter variables

Output parameters: 0

Output methodology: specifies whether to write output as a 1-D array containing only land points or as a 2-D array containing both land and water points.
1-d tile space includes the subgrid tiles and ensembles. 1-d grid space includes a vectorized, land-only grid-averaged set of values. Acceptable values are:

Value	Description
0	Do not write output
1	Write output in a 1-D tile domain
2	Write output in a 2-D grid domain
3	Write output in a 1-D grid domain

Output methodology: 2

Output data format: specifies the format of the model output data. Acceptable values are:

Value	Description
1	Write output in binary format
2	Write output in Grib format
3	Write output in NETCDF format

Output data format: 1

Output naming style: specifies the style of the model output names and their organization. Acceptable values are:

Value	Description
1	5 levels of hierarchy
2	3 levels of hierarchy
3	2 levels of hierarchy
4	WMO convention for weather codes

Output naming style: 1

Output GRIB Table Version: specifies GRIB table version.

Output GRIB Table Version: 130

Output GRIB Center Id: specifies GRIB center id.

Output GRIB Center Id: 7

Output GRIB Subcenter Id: specifies GRIB sub-center id.

Output GRIB Subcenter Id: 138

Output GRIB Process Id: specifies GRIB process id.

Output GRIB Process Id: 1

Output GRIB Grid Id: specifies GRIB grid id.

Output GRIB Grid Id: 255

Logging level: specifies different levels of logging. Acceptable values are:

Value	Description
1	Default messages
2	Default messages and additional messages
3	Debugging messages

Logging level: 1

Start mode: specifies if a restart mode is being used. Acceptable values are:

Value	Description
1	A restart mode is being used
2	A cold start mode is being used, no restart file read

When the cold start option is specified, the program is initialized using the LSM-specific initial conditions (typically assumed uniform for all tiles). When a restart mode is used, it is assumed that a corresponding restart file is provided depending upon which LSM is used. The user also needs to make sure that the ending time of the simulation is greater than model time when the restart file was written.

Start mode: 2

The start time is specified in the following format:

Variable	Value	Description
Starting year:	integer 2001 – present	specifying starting year
Starting month:	integer 1 – 12	specifying starting month
Starting day:	integer 1 – 31	specifying starting day
Starting hour:	integer 0 – 23	specifying starting hour
Starting minute:	integer 0 – 59	specifying starting minute
Starting second:	integer 0 – 59	specifying starting second

Starting year:	2002
Starting month:	10
Starting day:	29
Starting hour:	1
Starting minute:	0
Starting second:	0

The end time is specified in the following format:

Variable	Value	Description
Ending year:	integer 2001 – present	specifying ending year
Ending month:	integer 1 – 12	specifying ending month
Ending day:	integer 1 – 31	specifying ending day
Ending hour:	integer 0 – 23	specifying ending hour
Ending minute:	integer 0 – 59	specifying ending minute
Ending second:	integer 0 – 59	specifying ending second

Ending year: 2002
Ending month: 10
Ending day: 31
Ending hour: 1
Ending minute: 0
Ending second: 0

Model timestep: specifies the time-step for the run. Acceptable values are:

Value	Description
900	15 minute time-step
1800	30 minute time-step
3600	60 minute time-step

For a nested domain, the timesteps for each nest should be specified with white spaces as the delimiter. If two domains (one subnest) are employed, the first one using 900 seconds and the second one using 3600 seconds as the timestep, the model timesteps are specified as:

E.g.: Model timestep: 900 3600

Model timestep: 1800

Undefined value: specifies the undefined value. The default is set to -9999.

Undefined value: -9999

Output directory: specifies the name of the top-level output directory. Acceptable values are any 40 character string. The default value is set to OUTPUT. For simplicity, throughout the rest of this document, this top-level output directory shall be referred to by its default name, \$WORKING/LIS/OUTPUT.

```
Output directory:          'OUTPUT'
```

Diagnostic output file: specifies the name of run time diagnostic file. Acceptable values are any 40 character string.

```
Diagnostic output file:      'lisdiag'
```

Number of ensembles per tile: specifies the number of ensembles of tiles to be used. The value should be greater than equal to 1.

```
Number of ensembles per tile: 1
```

The following options are used for subgrid tiling based on vegetation

Maximum number of tiles per grid: defines the maximum tiles per grid (this can be as many as the total number of vegetation types).

```
Maximum number of tiles per grid: 1
```

Cutoff percentage: defines the smallest percentage of a cell for which to create a tile. The percentage value is expressed as a fraction.

```
Cutoff percentage: 0.05
```

This section specifies the 2-d layout of the processors in a parallel processing environment. The user can specify the number of processors along the east-west dimension and north-south dimension using **Number of processors along x:** and **Number of processors along y:**, respectively. Note that the layout of processors should match the total number of processors used. For example, if 8 processors are used, the layout can be specified as 1x8, 2x4, 4x2, or 8x1. Further, this section also allows the specification of halos around the domains on each processor using **Halo size along x:** and **Halo size along y:**.

```
Number of processors along x: 2
```

```
Number of processors along y:      2
Halo size along x: 0
Halo size along y: 0
```

9.4 Data assimilation

This section specifies the choice of data assimilation options.

Number of data assimilation instances: specifies the number of data assimilation instances. Valid values are 0 (no assimilation) or higher.

```
Number of data assimilation instances: 0
```

Assimilation algorithm: specifies the choice of data assimilation algorithms. Acceptable values are:

Value	Description
0	None
1	Direct Insertion
2	GMAO EnKF data assimilation

```
Assimilation algorithm: 0
```

Assimilation set: specifies the “assimilation set”, which is the instance related to the assimilation of a particular observation.

Acceptable values are:

Value	Description
1	Synthetic soil moisture (from Noah)
2	Synthetic soil moisture (from Catchment)
3	Synthetic SWE (from Noah)
8	Synthetic LST (from Noah)
7	ISCCP LST
9	AMSRE L3 soil moisture daily gridded data (HDF format)
10	MODIS sca
11	AFWA SNODEP
12	Synthetic multi-soil moisture observation types
13	NESDIS AMSRE soil moisture (binary format)
14	NASA AMSRE soil moisture (binary format)
15	ANSA snow data

Assimilation set: 0

Bias estimation

Bias estimation algorithm: specifies the dynamic bias estimation algorithm to use. Acceptable values are:

Value	Description
0	No dynamic bias estimation
1	NASA GMAO dynamic bias estimation

Bias estimation algorithm: 0

Exclude analysis increments: specifies if the analysis increments are to be skipped. This option is typically used along with the dynamic bias estimation algorithm. The user can choose to apply only the bias increments or both the bias increments and analysis increments. Acceptable values are:

Value	Description
0	Apply analysis increments
1	Do not apply analysis increments

Exclude analysis increments: 0

Use bias restart: This option specifies if the bias parameters are to be read from a checkpoint file. Acceptable values are:

Value	Description
0	Use a bias restart file
1	Do not use a bias restart file

Use bias restart: 0

Bias restart output frequency: Specifies the frequency (in seconds) of bias restart files

Bias restart output frequency: 86400

Number of state variables: specifies the number of state prognostic variables used in the assimilation.

Number of state variables: 0

Number of observation types: specifies the number of operation species/types used in the assimilation.

Number of observation types: 0

Output interval for diagnostics: specifies the output diagnostics interval in seconds.

Output interval for diagnostics: 21600

Write ensemble members: specifies if a binary output of ensemble members is to be written. Acceptable values are:

Value	Description
0	Do not write ensemble members
1	Write ensemble members

Write ensemble members: 0

Write observations: specifies if the processed, quality-controlled observations are to be written (Note that a corresponding observation plugin routine needs to be implemented) Acceptable values are:

Value	Description
0	Do not write observations
1	Write observations

Write observations: 0

Write innovations: specifies if a binary output of the normalized innovations is to be written. Acceptable values are:

Value	Description
0	Do not write innovations
1	Write innovations

Write innovations: 0

Perturbation options

Forcing perturbation algorithm: specifies the algorithm for perturbing the forcing variables

Acceptable values are:

Value	Description
0	None
1	Static random gaussian perturbations for all forcing variables except precip and Static random lognormal perturbations for precip
2	GMAO perturbation algorithm

Forcing perturbation algorithm: 0

Observation perturbation algorithm: specifies the algorithm for perturbing the observations. Acceptable values are:

Value	Description
0	None
1	Static random gaussian perturbations
2	GMAO perturbation algorithm

Observation perturbation algorithm: 0

State perturbation algorithm: specifies the algorithm for perturbing the state prognostic variables Acceptable values are:

Value	Description
0	None
1	Static random gaussian perturbations
2	GMAO perturbation algorithm

State perturbation algorithm: 0

Forcing perturbation frequency: specifies the forcing perturbation interval in seconds

Forcing perturbation frequency: 3600

Observation perturbation frequency: specifies the observation perturbation interval in seconds

Observation perturbation frequency: 3600

State perturbation frequency: specifies the prognostic variable perturbation interval in seconds

State perturbation frequency: 3600

Number of forcing fields to be perturbed: Specifies the number of forcing fields to be perturbed.

Number of forcing fields to be perturbed: 0

Forcing attributes file: ASCII file that specifies the attributes of the forcing. Please refer to the sample forcing_attribs.txt (Section 11.1) file for a complete specification description.

Forcing attributes file:

Forcing perturbation attributes file: ASCII file that specifies the attributes of the forcing perturbations. Please refer to the sample forcing_pertattribs.txt (Section 11.1) file for a complete specification description.

Forcing perturbation attributes file:

State attributes file: ASCII file that specifies the attributes of the prognostic variables. Please refer to the sample state_attribs.txt (Section 11.2) file for a complete specification description.

State attributes file:

State perturbation attributes file: ASCII file that specifies the attributes of the prognostic variable perturbations. Please refer to the sample state_pertattribs.txt (Section 11.2) file for a complete specification description.

State perturbation attributes file:

Observation attributes file: ASCII file that specifies the attributes of the observation variables. Please refer to the sample obs_attribs.txt (Section 11.3) file for a complete specification description.

Observation attributes file:

Observation perturbation attributes file: ASCII file that specifies the attributes of the observation variable perturbations. Please refer to the sample obs_pertattribs.txt (Section 11.3) file for a complete specification description.

Observation perturbation attributes file:

Bias estimation attributes file: ASCII file that specifies the attributes of the bias estimation. Please refer to the sample bias_attribs.txt (Section 11.4) file for a complete specification description.

Bias estimation attributes file:

Bias restart file: Specifies the restart file to be used for initializing bias parameters

Bias restart file:

Synthetic Soil Moisture Assimilation

Synthetic Soil Moisture data directory: specifies the directory for the synthetic soil moisture data.

Synthetic Soil Moisture data directory: ./input/dainput/SynSM/

Synthetic Soil Moisture (multiple observation types) Assimilation

Synthetic Soil Moisture data directory: specifies the directory for the synthetic soil moisture data.

Synthetic Soil Moisture data directory: ./input/dainput/SynSM/

Synthetic SWE Assimilation

Synthetic SWE data directory: specifies the directory for the synthetic snow water equivalent data.

Synthetic SWE data directory: ./input/dainput/SynSWE/

AMSR-E smc Assimilation

AMSR-E Soil Moisture data directory: specifies the directory for the AMSR-E soil moisture data.

AMSR-E and Noah CDF file directory: specifies the directory for the AMSR-E and Noah CDF file.

CDF levels: specifies the number of CDF levels.

AMSR-E Soil Moisture data directory: 'input'

AMSR-E and Noah CDF file directory: 'input'

CDF levels: 20

SNODEP Assimilation

SNODEP data directory: specifies the directory for the SNODEP data.

SNODEP mesh resolution: specifies the resolution of the SNODEP mesh (8 or 16).

SNODEP data directory: ./FORCING/AFWA1

SNODEP mesh resolution: 8

ISCCP Tskin Assimilation

ISCCP Tskin data directory: specifies the directory for the International Satellite Cloud Climatology Project (ISCCP) LST data.

ISCCP Tskin scale data: specifies if the LST observations are to be scaled or not (0 - no scaling, 1 - scale data)

ISCCP Tskin model mean data file: specifies the name of the file containing the monthly mean values from the LSM

ISCCP Tskin model std data file: specifies the name of the file containing the monthly standard deviation values from the LSM

ISCCP Tskin obs mean data file: specifies the name of the file containing the monthly mean values of the observations

ISCCP Tskin model std data file: specifies the name of the file containing the monthly standard deviation values of the observations

ISCCP Tskin data directory: '.../ISCCP/'

ISCCP Tskin scale data: 0

ISCCP Tskin model mean data file: .../SND_Input/noah_mean

ISCCP Tskin model std data file: .../SND_Input/noah_std

ISCCP Tskin obs mean data file: .../SND_Input/isccp_obs_mean

```
ISCCP Tskin obs std data file: ../SND_Input/isccp_obs_std
```

Synthetic LST Assimilation

Synthetic LST data directory: specifies the directory for the synthetic land surface temperature data.

Synthetic LST data directory:

MODIS snow cover fraction assimilation

MODIS SCF data directory: specifies the directory for the MODIS snow cover fraction data.

MODIS SCF use gap filled product: specifies whether the gap-filled product is to be used (1-use, 0-do not use)

MODIS SCF cloud threshold: Cloud cover threshold to be used for screening observations (in percentage)

MODIS SCF cloud persistence threshold: Cloud cover persistence threshold to be used for screening observations (in days)

```
MODIS SCF data directory: ./MODIS  
MODIS SCF use gap filled product: 1  
MODIS SCF cloud threshold: 90  
MODIS SCF cloud persistence threshold: 3
```

NESDIS AMSRE soil moisture Assimilation

NESDIS AMSRE soil moisture data directory: specifies the directory for the NESDIS AMSRE soil moisture data.

NESDIS AMSRE scale observations: specifies whether to scale the soil moisture data

NESDIS AMSRE model CDF file: specifies the model CDF file

NESDIS AMSRE observation CDF file: specifies the observation CDF file

```
NESDIS AMSRE soil moisture data directory: ./NESDIS_AMSRE/  
NESDIS AMSRE scale observations: 1  
NESDIS AMSRE model CDF file: ./noah_nldas_cdfs.bin  
NESDIS AMSRE observation CDF file: ./noaa_cdfs.bin
```

NASA AMSRE soil moisture Assimilation

NASA AMSRE soil moisture data directory: specifies the directory for the NASA AMSRE soil moisture data.
NASA AMSRE scale observations: specifies whether to scale the soil moisture data
NASA AMSRE model CDF file: specifies the model CDF file
NASA AMSRE observation CDF file: specifies the observation CDF file

NASA AMSRE soil moisture data directory: ./NASA_AMSRE/
NASA AMSRE scale observations: 1
NASA AMSRE model CDF file: ./noah_nldas_cdfs.bin
NASA AMSRE observation CDF file: ./nasa_cdfs.bin

ANSA snow assimilation

ANSA data directory: specifies the directory for the ANSA snow directory

ANSA snow directory:

9.5 Radiative Transfer/Forward Models

This section specifies the choice of radiative transfer or forward modeling tools
Radiative Transfer Model: specifies which RTM is to be used Acceptable values are:

Value	Description
1	CRTM

RTM invocation frequency: specifies (in seconds) the invocation frequency of the chosen RTM
RTM history output frequency: specifies (in seconds) the history output frequency of the RTM

Radiative Transfer Model: 1
RTM invocation frequency: 3600
RTM history output frequency: 10800

9.5.1 CRTM

This section specifies the specifications to enable a CRTM instance

CRTM number of sensors: specifies the number of sensors to be used
CRTM number of layers: specifies the number of atmospheric layers
CRTM number of absorbers: specifies the number of absorbers
CRTM number of clouds: specifies the number of cloud types
CRTM number of aerosols: specifies the number of aerosol types
CRTM sensor id: specifies the name of sensors to be simulated
CRTM coefficient data path: specifies the location of the files containing the CRTM coefficient data

```
CRTM number of sensors:      1
CRTM number of layers:       64
CRTM number of absorbers:    2
CRTM number of clouds:       0
CRTM number of aerosols:     0
CRTM sensor id:             amsua_n18
CRTM coefficient data path: ./Coefficient_Data/
```

9.6 Optimization

This section specifies the choice optimization tools

Optimization Algorithm: Specifies which algorithm is to be used for optimization Acceptable values are:

Value	Description
0	no optimization
1	Levenberg-Marquardt
2	Genetic Algorithm
3	SCE-UA Algorithm

Objective Function Method: specifies the objective function method Acceptable values are:

Value	Description
1	Least Squares

Optimization Algorithm: 2
Objective Function Method: 1

9.7 Parameter Estimation

This section provides specifications of parameter estimation instances

Optimization Set: specifies the “optimization set”, which represents the observation source used in the particular parameter estimation instance

Write PE Observations: specifies whether to output processed observations for parameter estimation

Optimization Set:

Write PE Observations:

2

1

9.7.1 Genetic Algorithm

This section provides specifications of the genetic algorithm instance

GA decision space attributes file: ASCII file that specifies the attributes of a decision space

Please refer to the sample decspace_attribs.txt (Section 12.1) file for a complete specification description.

GA restart file: specifies the name of the GA restart file

GA number of generations: specifies the number of generations of GA

GA number of children per parent: specifies how many offsprings are produced by two parent solutions (1 or 2)

GA crossover scheme: specifies the type of crossover scheme.

Acceptable values are:

Value	Description
1	single point crossover
2	uniform crossover

GA crossover probability: threshold to be used for conducting a crossover operation

GA mutation scheme: specifies the type of mutation scheme.

Acceptable values are:

Value	Description
0	jump mutation
1	creep mutation

GA creep mutation probability: specifies the creep mutation max threshold
GA jump mutation probability: specifies the jump mutation max threshold
GA use elitism: specifies whether to enable elitism in the selection of new solutions (0 - do not use, 1- use)
GA start mode: specifies the start mode (1 - restart from a file, 2 - cold start)

GA decision space attributes file:	./noah_sm_decspace.txt
GA restart file:	./OUTPUT/EXP999/GA/GA.188.GArst
GA number of generations:	100
GA number of children per parent:	1
GA crossover scheme:	2
GA crossover probability:	0.5
GA use creep mutations:	0
GA creep mutation probability:	0.04
GA jump mutation probability:	0.02
GA use elitism:	1
GA start mode:	2

9.7.2 Observations for Parameter Estimation

This section of the config file includes the observation specifications for parameter estimation

9.7.3 Walnut Gulch PBMR soil moisture data

WG PBMR soil moisture data directory: specifies the location of the Walnut Gulch PBMR soil moisture data
WG PBMR observations attributes file: specifies the location of the observation attributes file
WB PBMR site index: specifies the site location of the forcing data

WG PBMR soil moisture data directory:	../WG_domain/PBMR/
WG PBMR observations attributes file:	./wgPBMRsm_attribs.txt
WG PBMR site index:	5

9.8 Domain specification

This section of the config file specifies the running domain (domain over which the simulation is carried out). The specification of the running domain section depends on the type of LIS domain and projection used. Section 9.1 lists the projections that LIS supports.

```
\subsubsection{Cylindrical lat/lon} \label{sssec:run_latlon}
This section describes how to specify a cylindrical latitude/longitude
projection.
See Appendix~\ref{sec:d_latlon_example} for more details about
setting these values.
\begin{verbatim}

run domain lower left lat:          25.875
run domain lower left lon:          -124.875
run domain upper right lat:         52.875
run domain upper right lon:         -67.875
run domain resolution (dx):        0.25
run domain resolution (dy):        0.25
```

9.8.1 Polar stereographic

This section describes how to specify a polar stereographic projection. See Appendix B for more details about setting these values.

```
run domain lower left lat:          32.875
run domain lower left lon:          -104.875
run domain true lat:               36.875
run domain standard lon:           -98.875
run domain orientation:            0.0
run domain resolution:             25.0
run domain x-dimension size:      40
run domain y-dimension size:      30
```

9.8.2 Lambert conformal

This section describes how to specify a Lambert conformal projection. See Appendix D for more details about setting these values.

```
run domain lower left lat:           32.875
run domain lower left lon:          -104.875
run domain true lat1:                36.875
run domain true lat2:                36.875
run domain standard lon:            -98.875
run domain resolution:               25.0
run domain x-dimension size:        40
run domain y-dimension size:        30
```

9.8.3 Mercator

This section describes how to specify a Mercator projection. See Appendix E for more details about setting these values.

```
run domain lower left lat:           32.875
run domain lower left lon:          -104.875
run domain true lat1:                36.875
run domain standard lon:            -98.875
run domain resolution:               25.0
run domain x-dimension size:        40
run domain y-dimension size:        30
```

9.8.4 Gaussian

This section describes how to specify a Gaussian projection. See Appendix C for more details about setting these values.

```
run domain first grid point lat:     -89.27665
run domain first grid point lon:      0.0
run domain last grid point lat:      89.27665
run domain last grid point lon:     -0.9375
run domain resolution dlon:          0.9375
run domain number of lat circles:    95
```

9.8.5 UTM

This section describes how to specify a UTM projection. See Appendix F for more details about setting these values.

```
run domain UTM zone:           12
run domain northing of SW corner: 3507393.0
run domain easting of SW corner: 586018.0
run domain x-dimension size:    660
run domain y-dimension size:    333
run domain resolution:          40

#Definition of Parameter Domain
```

9.8.6 Cylindrical lat/lon

This section describes how to specify a cylindrical latitude/longitude projection. See Appendix A for more details about setting these values.

```
param domain lower left lat:      -59.875
param domain lower left lon:       -179.875
param domain upper right lat:     89.875
param domain upper right lon:     179.875
param domain resolution (dx):     0.25
param domain resolution (dy):     0.25
```

9.8.7 Gaussian

This section describes how to specify a Gaussian projection. See Appendix C for more details about setting these values.

```
param domain first grid point lat: -89.27665
param domain first grid point lon:  0.0
param domain last grid point lat:  89.27665
param domain last grid point lon: -0.9375
param domain resolution dlon:     0.9375
param domain number of lat circles: 95
```

9.8.8 UTM

This section describes how to specify a UTM projection. See Appendix F for more details about setting these values.

```
param domain UTM zone:           12
param domain northing of SW corner: 3507393.0
param domain easting of SW corner: 586018.0
param domain x-dimension size:    660
param domain y-dimension size:    333
param domain resolution:          40
```

9.9 Parameters

```
#Metadata for Parameter maps
#Landcover and Landmask
```

landmask file: specifies the location of land/water mask file.
landcover file: specifies the location of the vegetation classification file
landcover file format: specifies if the vegetation file is tiled or not (0-not tiled, 1- tiled)

```
landmask file:                  ./input/UMD-25KM/UMD_mask0.25.1gd4r
landcover file:                 ./input/UMD-25KM/UMD_veg0.25.1gd4r
landcover file format:          1
```

This section should also specify the domain specifications of the landcover data. If the map projection of parameter data is specified to be lat/lon, the following configuration should be used for specifying landcover data See Appendix A for more details about setting these values.

```
landcover lower left lat:        -59.875
landcover lower left lon:        -179.875
landcover upper right lat:       89.875
landcover upper right lon:       179.875
landcover resolution (dx):      0.25
landcover resolution (dy):      0.25
```

If the map projection of parameter data is specified to be gaussian, the following configuration should be used for specifying landcover data See Appendix C for more details about setting these values.

```
landcover first grid point lat:  
landcover first grid point lon:  
landcover last grid point lat:  
landcover last grid point lon:  
landcover resolution dlon:  
landcover number of lat circles:
```

If the map projection of parameter data is specified to be polar stereographic, the following configuration should be used for specifying landcover data See Appendix B for more details about setting these values.

```
landcover lower left lat:  
landcover lower left lon:  
landcover true lat:  
landcover standard lon:  
landcover orientation:  
landcover resolution:  
landcover x-dimension size:  
landcover y-dimension size:
```

If the map projection of parameter data is specified to be UTM, the following configuration should be used for specifying landcover data See Appendix F for more details about setting these values.

```
landcover UTM zone: 12  
landcover northing of SW corner: 3507393.0  
landcover easting of SW corner: 586018.0  
landcover x-dimension size: 660  
landcover y-dimension size: 333  
landcover resolution: 40
```

Topography maps **elevation map**: specifies the elevation of the LIS grid.

slope map: specifies the slope of the LIS grid.

aspect map: specifies the aspect of the LIS grid.

curvature map: specifies the curvature of the LIS grid.

```
elevation map: ./input/UMD-25KM/lis_elev.1gd4r
```

```
slope map:  
aspect map:  
curvature map:
```

This section should also specify the domain specifications of the topography data. If the map projection of parameter data is specified to be lat/lon, the following configuration should be used for specifying topography data See Appendix A for more details about setting these values.

```
topography lower left lat:      -59.875  
topography lower left lon:      -179.875  
topography upper right lat:     89.875  
topography upper right lon:     179.875  
topography resolution (dx):    0.25  
topography resolution (dy):    0.25
```

If the map projection of parameter data is specified to be gaussian, the following configuration should be used for specifying topography data See Appendix C for more details about setting these values.

```
topography first grid point lat: -89.27665  
topography first grid point lon:  0.0  
topography last grid point lat:  89.27665  
topography last grid point lon: -0.9375  
topography resolution dlon:     0.9375  
topography number of lat circles: 95
```

If the map projection of parameter data is specified to be polar stereographic, the following configuration should be used for specifying topography data See Appendix B for more details about setting these values.

```
topography lower left lat:  
topography lower left lon:  
topography true lat:  
topography standard lon:  
topography orientation:  
topography resolution:  
topography x-dimension size:  
topography y-dimension size:
```

If the map projection of parameter data is specified to be UTM, the following configuration should be used for specifying topography data See Appendix F for more details about setting these values.

```
topography UTM zone:           12
topography northing of SW corner: 3507393.0
topography easting of SW corner: 586018.0
topography x-dimension size:    660
topography y-dimension size:    333
topography resolution:          40
```

Soils maps
soil texture map: specifies the soil texture file.
sand fraction map: specifies the sand fraction map file.
clay fraction map: specifies the clay fraction map file.
silt fraction map: specifies the silt map file.
soil color map: specifies the soil color map file.
porosity map: specifies porosity.
saturated matric potential map: specifies saturated matric potential
saturated hydraulic conductivity map: specifies saturated hydraulic conductivity
b parameter map: specifies b parameter
quartz map: specifies quartz data

```
soil texture map:
sand fraction map:           ./input/UMD-25KM/sandfao.1gd4r
clay fraction map:            ./input/UMD-25KM/clayfao.1gd4r
silt fraction map:            ./input/UMD-25KM/siltfao.1gd4r
soil color map:
porosity map:
saturated matric potential map: # psisat
saturated hydraulic conductivity map: # ksat
b parameter map:
quartz map:
```

If the map projection of parameter data is specified to be lat/lon, the following configuration should be used for specifying soils data See Appendix A for more details about setting these values.

```
soils lower left lat:          -59.875
soils lower left lon:          -179.875
soils upper right lat:         89.875
soils upper right lon:         179.875
soils resolution (dx):        0.25
soils resolution (dy):        0.25
```

If the map projection of parameter data is specified to be gaussian, the following configuration should be used for specifying soils data See Appendix C for more details about setting these values.

```
soils domain first grid point lat:      -89.27665
soils domain first grid point lon:       0.0
soils domain last grid point lat:        89.27665
soils domain last grid point lon:        -0.9375
soils domain resolution dlon:           0.9375
soils domain number of lat circles:     95
```

If the map projection of parameter data is specified to be polar stereographic, the following configuration should be used for specifying soils data See Appendix B for more details about setting these values.

```
soils lower left lat:
soils lower left lon:
soils true lat:
soils standard lon:
soils orientation:
soils resolution:
soils x-dimension size:
soils y-dimension size:
```

If the map projection of parameter data is specified to be UTM, the following configuration should be used for specifying soils data See Appendix F for more details about setting these values.

```
soils UTM zone:                12
soils northing of SW corner:   3507393.0
soils easting of SW corner:    586018.0
```

```
soils x-dimension size: 660
soils y-dimension size: 333
soils resolution: 40
```

Albedo maps **albedo map**: specifies the source of the climatology based albedo files. The climatology albedo data files have the following naming convention: `./directory/_file header_.itag_.1gd4r` The tag should be either sum, win, spr, or aut depending on the season. The file header can be anything (such as alb1KM). **albedo climatology interval**: specifies the frequency of the albedo climatology in months.

max snow free albedo map: specifies the static upper bound of snow free albedo file.

```
albedo map: ./input/UMD-25KM/alb
albedo climatology interval: 3
max snow free albedo map: ./input/UMD-25KM/global_mxsnoalb.25km.1gd4r
```

bottom temperature map: specifies the bottom boundary temperature data.

```
bottom temperature map: ./input/UMD-25KM/tbot_0.25.1gd4r
```

Greenness fraction maps **greenness fraction map**: specifies the source of the climatology based gfrac files. The climatology greenness data files have the following naming convention: `./directory/_file header_.itag_.1gd4r` The tag should represent the month (such as jan, feb, mar, etc.). The file header can be anything (such as green1KM).

greenness climatology interval: specifies the frequency of the greenness climatology in months.

```
greenness fraction map: ./input/UMD-25KM/green
greenness climatology interval: 1
```

LAI/SAI maps **LAI map**: specifies the source of the LAI climatology files The climatology LAI data files have the following naming convention: `./directory/_file header_.itag_.1gd4r` The tag should represent the month (such as jan, feb, mar,

etc.). The file header can be anything (such as LAI1KM).
SAI map: specifies the source of the SAI climatology files. The climatology SAI data files have the following naming convention: `./directory/_/file header._.tag._.1gd4r`. The tag should represent the month (such as jan, feb, mar, etc.). The file header can be anything (such as SAI1KM).

LAI map:	<code>./input/UMD-25KM/AVHRR_LAI_CLIM</code>
SAI map:	<code>./input/UMD-25KM/AVHRR_SAI_CLIM</code>

snow depth maps **Snow depth map:** specifies the location of the realtime snow depth.

Snow depth map:	<code>./FORCING/AFWA1</code>
-----------------	------------------------------

9.10 Forcings

9.10.1 GDAS

GDAS forcing directory: specifies the location of the GDAS forcing files.
GDAS T126 elevation map: specifies the GDAS T126 elevation definition.
GDAS T170 elevation map: specifies the GDAS T170 elevation definition.
GDAS T254 elevation map: specifies the GDAS T254 elevation definition.
GDAS T382 elevation map: specifies the GDAS T382 elevation definition.
GDAS domain x-dimension size: specifies the number of columns of the native domain parameters of the GDAS forcing data. The map projection is specified in the driver modules defined for the GDAS routines.
GDAS domain y-dimension size: specifies the number of rows of the native domain parameters of the GDAS forcing data. The map projection is specified in the driver modules defined for the GDAS routines.
GDAS number of forcing variables: specifies the number of forcing variables provided by GDAS at the model initialization step.

GDAS forcing directory:	<code>./input/FORCING/GDAS/</code>
GDAS T126 elevation map:	
GDAS T170 elevation map:	<code>./input/UMD-25KM/gdas_T170_elev.1gd4r</code>
GDAS T254 elevation map:	<code>./input/UMD-25KM/gdas_T254_elev.1gd4r</code>
GDAS T382 elevation map:	<code>./input/UMD-25KM/gdas_T382_elev.1gd4r</code>

```
GDAS domain x-dimension size:      512
GDAS domain y-dimension size:      256
GDAS number of forcing variables:  10
```

9.10.2 GEOS

GEOS forcing directory: specifies the location of the GEOS forcing files.
GEOS domain x-dimension size: specifies the number of columns of the native domain parameters of the GEOS forcing data. The map projection is specified in the driver modules defined for the GEOS routines.
GEOS domain y-dimension size: specifies the number of rows of the native domain parameters of the GEOS forcing data. The map projection is specified in the driver modules defined for the GEOS routines.
GEOS number of forcing variables: specifies the number of forcing variables provided by GEOS at the model initialization step.

```
GEOS forcing directory:          ./input/FORCING/GEOS/BEST_LK/
GEOS domain x-dimension size:   360
GEOS domain y-dimension size:   181
GEOS number of forcing variables: 13
```

9.10.3 ECMWF

ECMWF forcing directory: specifies the location of the ECMWF forcing files.//
ECMWF elevation map 0: specifies the ECMWF elevation definition.// **ECMWF elevation map 1:** specifies the ECMWF elevation definition.// **ECMWF elevation map 2:** specifies the ECMWF elevation definition.// **ECMWF domain x-dimension size:** specifies the number of columns of the native domain parameters of the ECMWF forcing data. The map projection is specified in the driver modules defined for the ECMWF routines. // **ECMWF domain y-dimension size:** specifies the number of rows of the native domain parameters of the ECMWF forcing data. The map projection is specified in the driver modules defined for the ECMWF routines. // **ECMWF number of forcing variables:** specifies the number of forcing variables provided by ECMWF at the model initialization step.

```
ECMWF forcing directory:          ./input/FORCING/ECMWF/
ECMWF elevation map 0:
ECMWF elevation map 1:
ECMWF elevation map 2:
ECMWF domain x-dimension size:
ECMWF domain y-dimension size:
ECMWF number of forcing variables:
```

9.10.4 ECMWF Reanalysis

ECMWF Reanalysis forcing directory: specifies the location of the ECMWF Reanalysis forcing files.// **ECMWF Reanalysis maskfile:** specifies the file containing the ECMWF Reanalysis land/sea mask.// **ECMWF Reanalysis elevation map:** specifies the file containing the ECMWF Reanalysis elevation definition.// **ECMWF Reanalysis domain x-dimension size:** specifies the number of columns of the ECMWF Reanalysis domain.// **ECMWF Reanalysis domain y-dimension size:** specifies the number of rows of the ECMWF Reanalysis domain.//

```
ECMWF Reanalysis forcing directory:      ./input/FORCING/ECMWF-REANALYSIS/
ECMWF Reanalysis maskfile:             ./input/FORCING/ECMWF-REANALYSIS/ecmwf_land_sea.05
ECMWF Reanalysis elevation map:       ./input/UMD-25KM/ecmwf_reanal_elev.1gd4r
ECMWF Reanalysis domain x-dimension size:    720
ECMWF Reanalysis domain y-dimension size:    360
```

9.10.5 AFWA/AGRMET

AGRMET forcing directory: specifies the location of the root directory containing the input files. The AGRMET processing algorithms assumes the following hierarchy under the root directory at each instance. For example, if the root directory for storing the files is 'FORCING/AFWA/', and the current instance is decemeber 1st, 2005, then the files are stored under the 'FORCING/AFWA/20051201/' directory. The additional directory names specified here are assumed to be located under this root directory.// **AGRMET analysis directory:** specifies the location where temporary precip analysis fields will

be written// AGRMET surface fields directory: specifies the location of the surface fields (sfc*).// AGRMET merged precip directory: specifies the location of the processed precip obs (presav_*).// AGRMET merged precip directory: specifies the location of the WWMCA data (WWMCA*).// AGRMET GFS data directory: specifies the location of the GFS data (MT.avn*).// AGRMET SSMI data directory: specifies the location of the SSM/I data (ssmira_*).// AGRMET GEOPRECIP data directory: specifies the location of the GEOPRECIP files (prec08* and rank08*).// AGRMET CDMS data directory: specifies the location of the surface and precip obs (sfcobs_* and preobs_*).// AGRMET use timestamp on directories: specifies whether or not to use a timestamp on directories. Acceptable values are:

Value	Description
0	do not use timestamp
1	use timestamp

AGRMET polar mask file: specifies the landmask in 8th mesh polar stereographic projection used by the AGRMET algorithms. // AGRMET polar terrain file: specifies the terrain in 8th mesh polar stereographic projection used by the AGRMET algorithms. // AGRMET sfcalc cntm file: specifies the name of the files with the spreading radii used for the barnes analysis on the GFS and surface obs. // AGRMET precip climatology: specifies the path to the precip climatology data.// AGRMET nogaps wind weight: specifies the weighting factor for the first guess winds// AGRMET minimum wind speed: specifies the minimum allowable wind speed on the AGRMET grid// AGRMET use present/past weather estimate: specifies whether to use present/past weather estimates. Acceptable values are:

Value	Description
0	do not use estimates
1	use estimates

AGRMET use precip observations: specifies whether to use precip observations. Acceptable values are:

Value	Description
0	do not use observations
1	use observations

AGRMET use SSMI data: specifies whether to use SSM/I data Acceptable values are:

Value	Description
0	do not use SSM/I data
1	use SSM/I data

AGRMET use CDFSII-based estimate: specifies whether to use a CDFS-II based estimate. Acceptable values are:

Value	Description
0	do not use estimate
1	use estimate

AGRMET use GEOPRECIP estimate: specifies whether to use a GEOPRECIP based estimate. Acceptable values are:

Value	Description
0	do not use
1	use the estimate, do not use the rank
2	use the estimate and use the rank

AGRMET CDFSII time interval: specifies the CDFS-II time interval to look for cloud amount. Current value is 6.

AGRMET use precip climatology: specifies whether to use precip climatology. Acceptable values are:

Value	Description
0	do not use precip climatology
1	use precip climatology

AGRMET SSMI zero use switch: specifies whether to use SSM/I zeros. Acceptable values are:

Value	Description
0	do not use zeros
1	use zeros

AGRMET snow distribution shape parameter: specifies the snow distribution shape parameter (A typical value is 2.6)

AGRMET alternate monthly weighting factor: specifies the alternate monthly weighting factor used in the precip processing.

AGRMET minimum 3hr climo value: specifies a minimum 3 hour precip climo value required to generate a non-zero CDFSII total cloud-based precip estimate. A typical value is 0.025.

AGRMET maximum 3hr climo value: specifies a maximum 3 hour precip climo value required to generate a non-zero CDFSII total cloud-based precip estimate. A typical value is 0.375.

AGRMET minimum precip-per-precip day multiplier: specifies a minimum precip-per-precip day multiplier used to generate a non-zero CDFSII total cloud based precip estimate. A typical value is 0.0.

AGRMET maximum precip-per-precip day multiplier: specifies a maximum precip-per-precip day multiplier used to generate a non-zero CDFSII total cloud based precip estimate. A typical value is 1.1.

AGRMET cloud threshold to generate CDFSII estimate: specifies the cloud threshold to generate a CDFSII-based estimate. A typical value is 85.0.

AGRMET median cloud cover percentage1: specifies the median cloud cover percentage to move to for the CDFSII based precip estimate. A typical value is 15.0.

AGRMET median cloud cover percentage2: specifies the median cloud cover percentage to move to for the CDFSII based precip estimate. A typical value is 0.60.

AGRMET overcast percentage: specifies the overcast percentage to move to for CDFSII based precipitation estimate. A typical value is 0.30.

AGRMET 3hr maximum precip ceiling: specifies the 3 hour maximum precip ceiling value. A typical value is 200.0.

AGRMET forcing directory:	./FORCING/
AGRMET analysis directory:	./Analysis
AGRMET surface fields directory:	SFCALC
AGRMET merged precip directory:	PRECIP
AGRMET cloud data directory:	WWMCA
AGRMET GFS data directory:	GFS
AGRMET SSMI data directory:	SSMI
AGRMET GEOPRECIP data directory:	GEO
AGRMET CDMS data directory:	CDMS
AGRMET use timestamp on directories:	1
AGRMET polar mask file:	./STATIC/point_switches
AGRMET polar terrain file:	./STATIC/terrain
AGRMET sfcalc cntm file:	./STATIC/sfcalc-cntm
AGRMET precip climatology:	./STATIC/pcp_clim/
AGRMET nogaps wind weight:	0.75
AGRMET minimum wind speed:	0.25
AGRMET use present/past weather estimate:	1
AGRMET use precip observations:	1
AGRMET use SSMI data:	1
AGRMET use CDFSII-based estimate:	1
AGRMET use GEOPRECIP estimate:	2
AGRMET CDFSII time interval:	6
AGRMET use precip climatology:	1
AGRMET SSMI zero use switch:	1
AGRMET snow distribution shape parameter:	2.6
AGRMET alternate monthly weighting factor:	1.0
AGRMET minimum 3hr climo value:	0.025
AGRMET maximum 3hr climo value:	0.375
AGRMET minimum precip-per-precip day multiplier:	0.0
AGRMET maximum precip-per-precip day multiplier:	1.1
AGRMET cloud threshold to generate CDFSII estimate:	85.0

```
AGRMET median cloud cover percentage1:      15.0
AGRMET median cloud cover percentage2:      0.60
AGRMET overcast percentage:                 0.30
AGRMET 3hr maximum precip ceiling:        200.0
```

9.10.6 PRINCETON

PRINCETON forcing directory: specifies the location of the PRINCETON forcing files.

PRINCETON elevation map: specifies the PRINCETON elevation definition.

PRINCETON domain x-dimension size: specifies the number of columns of the native domain parameters of the PRINCETON forcing data. The map projection is specified in the driver modules defined for the PRINCETON routines.

PRINCETON domain y-dimension size: specifies the number of rows of the native domain parameters of the PRINCETON forcing data. The map projection is specified in the driver modules defined for the PRINCETON routines.

PRINCETON number of forcing variables: specifies the number of forcing variables provided by PRINCETON at the model initialization step.

```
PRINCETON forcing directory:      ./input/FORCING/PRINCETON
PRINCETON elevation map:         ./input/UMD-100KM/hydro1k_elev_mean_1d.1gd4r
PRINCETON domain x-dimension size:   360
PRINCETON domain y-dimension size:   180
PRINCETON number of forcing variables: 7
```

9.10.7 Rhone AGG

Rhone AGG forcing directory: specifies the location of the Rhone AGG forcing files.

Rhone AGG domain x-dimension size: specifies the number of columns of the native domain parameters of the Rhone AGG forcing data. The map projection is specified in the driver modules defined for the Rhone AGG routines.

Rhone AGG domain y-dimension size: specifies the number of rows of the native domain parameters of the Rhone AGG forcing data. The map projection is specified in the driver modules defined for the Rhone AGG routines.

```
Rhone AGG forcing directory:      ./input/FORCING/RHONE
Rhone AGG domain x-dimension size: 5
Rhone AGG domain y-dimension size: 6
```

9.10.8 GSWP

GSWP landmask file: specifies the GSWP2 landmask file.
GSWP domain x-dimension size: specifies the number of columns of the GSWP2 domain.
GSWP domain y-dimension size: specifies the number of rows of the GSWP2 domain.
GSWP number of forcing variables: specifies the number of GSWP2 forcing variables.
GSWP 2m air temperature map: specifies the GSWP2 2 meter air temperature data.
GSWP 2m specific humidity map: specifies the GSWP2 2 meter specific humidity data.
GSWP wind map: specifies the GSWP2 wind data.
GSWP surface pressure map: specifies the GSWP2 surface pressure data.
GSWP convective rainfall rate map: specifies the GSWP2 convective rainfall rate data.
GSWP rainfall rate map: specifies the GSWP2 rainfall rate data.
GSWP snowfall rate map: specifies the GSWP2 snowfall rate data.
GSWP incident shortwave radiation map: specifies the GSWP2 incident shortwave radiation data.
GSWP incident longwave radiation map: specifies the GSWP2 incident longwave radiation data.

GSWP landmask file:	./input/gswp2data/Fixed/landmask_gswp.nc
GSWP domain x-dimension size:	360
GSWP domain y-dimension size:	150
GSWP number of forcing variables:	10
GSWP 2m air temperature map:	./input/gswp2data/Tair_cru/Tair_cru
GSWP 2m specific humidity map:	./input/gswp2data/Qair_cru/Qair_cru
GSWP wind map:	./input/gswp2data/Wind_ncep/Wind_ncep
GSWP surface pressure map:	./input/gswp2data/PSurf_ecor/PSurf_ecor
GSWP convective rainfall rate map:	./input/gswp2data/Rainf_C_gswp/Rainf_C_gswp
GSWP rainfall rate map:	./input/gswp2data/Rainf_gswp/Rainf_gswp
GSWP snowfall rate map:	./input/gswp2data/Snowf_gswp/Snowf_gswp

```
GSWP incident shortwave radiation map: ./input/gswp2data/SWdown_srb/SWdown_srb  
GSWP incident longwave radiation map: ./input/gswp2data/LWdown_srb/LWdown_srb
```

9.10.9 GMAO GLDAS

GMAO GLDAS forcing directory: specifies the location of the GMAO GLDAS forcing files.

GMAO GLDAS domain x-dimension size: specifies the number of columns of the native domain parameters of the GMAO GLDAS forcing data.

GMAO GLDAS domain y-dimension size: specifies the number of rows of the native domain parameters of the GMAO GLDAS forcing data.

GMAO GLDAS number of forcing variables: specifies the number of forcing variables provided in the GMAO GLDAS data

```
GMAOGLDAS forcing directory:      .../FORCING/GLDAS_GMAO/  
GMAOGLDAS domain x-dimension size: 144  
GMAOGLDAS domain y-dimension size: 76  
GMAOGLDAS number of forcing variables: 10
```

9.10.10 GFS

GFS forcing directory: specifies the location of the GFS forcing files.

GFS T126 elevation map: specifies the GFS T126 elevation definition.

GFS T170 elevation map: specifies the GFS T170 elevation definition.

GFS T254 elevation map: specifies the GFS T254 elevation definition.

GFS T382 elevation map: specifies the GFS T382 elevation definition.

GFS domain x-dimension size: specifies the number of columns of the native domain parameters of the GFS forcing data. The map projection is specified in the driver modules defined for the GFS routines.

GFS domain y-dimension size: specifies the number of rows of the native domain parameters of the GFS forcing data. The map projection is specified in the driver modules defined for the GFS routines.

GFS number of forcing variables: specifies the number of forcing variables provided by GFS at the model initialization step.

```
GFS forcing directory:      ./input/FORCING/GFS/  
GFS T126 elevation map:    ./input/UMD-25KM/gdas_T126_elev.1gd4r  
GFS T170 elevation map:    ./input/UMD-25KM/gdas_T170_elev.1gd4r
```

```
GFS T254 elevation map:          ./input/UMD-25KM/gdas_T254_elev.1gd4r
GFS T382 elevation map:          ./input/UMD-25KM/gdas_T382_elev.1gd4r
GFS domain x-dimension size:    512
GFS domain y-dimension size:    256
GFS number of forcing variables: 10
```

9.10.11 GSWP1

GSWP1 forcing directory: specifies the location of the GSWP1 forcing files.

GSWP1 domain x-dimension size: specifies the number of columns of the native domain parameters of the GSWP1 forcing data. The map projection is specified in the driver modules defined for the GSWP1 routines.

GSWP1 domain y-dimension size: specifies the number of rows of the native domain parameters of the GSWP1 forcing data. The map projection is specified in the driver modules defined for the GSWP1 routines.

GSWP1 number of forcing variables: specifies the number of forcing variables provided by GSWP1 at the model initialization step.

```
GSWP1 forcing directory:          ./input/FORCING/GSWP1
GSWP1 domain x-dimension size: 360
GSWP1 domain y-dimension size: 150
GSWP1 number of forcing variables: 9
```

9.11 Supplemental forcings

9.11.1 AGRMET radiation

AGRRAD forcing directory: specifies the directory containing AGRMET radiation data.

```
AGRRAD forcing directory:          ./input/FORCING/AGRRAD
```

9.11.2 AGRMET polar stereographic radiation

AGRRADPS forcing directory: specifies the directory containing AGRMET polar stereographic radiation data.

AGRRADPS forcing directory: ./input/FORCING/AGRRADPS

9.11.3 CMAP precipitation

CMAP forcing directory: specifies the location of the CMAP forcing files.

CMAP domain x-dimension size: specifies the number of columns of the native domain parameters of the CMAP forcing data. The map projection is specified in the driver modules defined for the CMAP routines.

CMAP domain y-dimension size: specifies the number of rows of the native domain parameters of the CMAP forcing data. The map projection is specified in the driver modules defined for the CMAP routines.

CMAP forcing directory: ./input/FORCING/CMAP

CMAP domain x-dimension size: 512

CMAP domain y-dimension size: 256

9.11.4 CEOP station data

CEOP station forcing – during EOP1

CEOP location index: specifies the location of the CEOP station.

CEOP forcing directory: specifies the location of the CEOP forcing files.

CEOP metadata file: specifies the file containing CEOP metadata.

CEOP location index: 3 #SGP location

CEOP forcing directory: ./input/FORCING/CEOP/sgp.cfr

CEOP metadata file: `./input/FORCING/CEOP/sgp.mdata`

9.11.5 SCAN station data

SCAN forcing directory: specifies the location of the SCAN forcing files.

SCAN metadata file: specifies the file containing SCAN metadata.

SCAN forcing directory: `./input/FORCING/SCAN`

SCAN metadata file: `./input/FORCING/SCAN/msu_scan.mdata`

9.11.6 NLDAS

NLDAS forcing directory: specifies the location of the NLDAS forcing files.

NLDAS domain x-dimension size: specifies the number of columns of the native domain parameters of the NLDAS forcing data. The map projection is specified in the driver modules defined for the NLDAS routines.

NLDAS domain y-dimension size: specifies the number of rows of the native domain parameters of the NLDAS forcing data. The map projection is specified in the driver modules defined for the NLDAS routines.

NLDAS elevation map: specifies the NLDAS elevation definition.

EDAS height map: specifies the EDAS height definition.

EDAS height lower left lat: specifies the lower left latitude of the EDAS domain. (cylindrical latitude/longitude projection)

EDAS height lower left lon: specifies the lower left longitude of the EDAS domain. (cylindrical latitude/longitude projection)

EDAS height upper right lat: specifies the upper right latitude of the EDAS domain. (cylindrical latitude/longitude projection)

EDAS height upper right lon: specifies the upper right longitude of the EDAS domain. (cylindrical latitude/longitude projection)

EDAS height resolution (dx): specifies the resolution of the of the EDAS domain along the east-west direction

EDAS height resolution (dy): specifies the resolution of the of the EDAS domain along the north-south direction

NLDAS forcing directory: `./input/FORCING/NLDAS`

NLDAS domain x-dimension size:	464
NLDAS domain y-dimension size:	224
NLDAS elevation difference map:	./input/FORCING/NLDAS/elevdiff.bin
EDAS height map:	./input/N0.125/edas_elev.1gd4r
EDAS height lower left lat:	25.0625
EDAS height lower left lon:	-124.9375
EDAS height upper right lat:	52.9375
EDAS height upper right lon:	-67.0625
EDAS height resolution (dx):	0.125
EDAS height resolution (dy):	0.125

NLDAS2 forcing directory: specifies the location of the NLDAS2 forcing files.

NLDAS2 elevation map: specifies the NLDAS2 elevation definition.

NLDAS2 domain x-dimension size: specifies the number of columns of the native domain parameters of the NLDAS2 forcing data. The map projection is specified in the driver modules defined for the NLDAS2 routines.

NLDAS2 domain y-dimension size: specifies the number of rows of the native domain parameters of the NLDAS2 forcing data. The map projection is specified in the driver modules defined for the NLDAS2 routines.

NLDAS2 use model level data: specifies whether or not to read in the model level data (instead of 2/10m fields) from the NLDAS2 forcing dataset (will open up and read "B" files)

Note that this will read in "Height of Atmospheric Forcing" and "Surface Exchange Coefficient for Heat". You must make sure that they are included in your forcing variables list file. Acceptable values are:

Value	Description
0	do not use
1	use

NLDAS2 use model based swdown: specifies whether or not to read in the un-bias corrected model downward shortwave radiation data (in lieu of the bias corrected data) from the NLDAS2 forcing dataset (will open up and read "B" files) Acceptable values are:

Value	Description
0	do not use
1	use

NLDAS2 use model based precip: specifies whether or not to read in the model base precipitation data (instead of the observation based precipitation) from the NLDAS2 forcing dataset (will open up and read "B" files) Acceptable values are:

Value	Description
-------	-------------

0	do not use
---	------------

1	use
---	-----

NLDAS2 forcing directory:	./input/FORCING/NLDAS2
NLDAS2 elevation map:	
NLDAS2 domain x-dimension size:	464
NLDAS2 domain y-dimension size:	224
NLDAS2 use model level data:	0
NLDAS2 use model based swdown:	0
NLDAS2 use model based precip:	0

9.11.7 SALDAS

SALDAS forcing directory: specifies the location of the SALDAS forcing files.

SALDAS elevation map: specifies the SALDAS elevation definition.

SALDAS elevation map change 1: specifies the SALDAS elevation change 1 definition.

SALDAS elevation map change 2: specifies the SALDAS elevation change 2 definition.

SALDAS elevation map change 3: specifies the SALDAS elevation change 3 definition.

SALDAS domain x-dimension size: specifies the number of columns of the native domain parameters of the SALDAS forcing data. The map projection is specified in the driver modules defined for the SALDAS routines.

SALDAS domain y-dimension size: specifies the number of rows of the native domain parameters of the SALDAS forcing data. The map projection is specified in the driver modules defined for the SALDAS routines.

SALDAS number of forcing variables: specifies the number of forcing variables provided by SALDAS at the model initialization step.

SALDAS forcing directory:	./input/FORCING/SALDAS
SALDAS elevation map:	
SALDAS elevation map change 1:	
SALDAS elevation map change 2:	
SALDAS elevation map change 3:	
SALDAS domain x-dimension size:	
SALDAS domain y-dimension size:	
SALDAS number of forcing variables:	

9.11.8 TRMM 3B42RT precipitation

TRMM 3B42RT forcing directory: specifies the location of the TRMM 3B42RT forcing files.

TRMM 3B42RT domain x-dimension size: specifies the number of columns of the native domain parameters of the TRMM 3B42RT forcing data. The map projection is specified in the driver modules defined for the TRMM 3B42RT routines.

TRMM 3B42RT domain y-dimension size: specifies the number of rows of the native domain parameters of the TRMM 3B42RT forcing data. The map projection is specified in the driver modules defined for the TRMM 3B42RT routines.

```
TRMM 3B42RT forcing directory:      ./input/FORCING/3B42RT/
```

```
TRMM 3B42RT domain x-dimension size: 1440
```

```
TRMM 3B42RT domain y-dimension size: 480
```

9.11.9 TRMM 3B42V6 precipitation

TRMM 3B42V6 forcing directory: specifies the location of the TRMM 3B42V6 forcing files.

TRMM 3B42V6 domain x-dimension size: specifies the number of columns of the native domain parameters of the TRMM 3B42V6 forcing data. The map projection is specified in the driver modules defined for the TRMM 3B42V6 routines.

TRMM 3B42V6 domain y-dimension size: specifies the number of rows of the native domain parameters of the TRMM 3B42V6 forcing data. The map projection is specified in the driver modules defined for the TRMM 3B42V6 routines.

```
TRMM 3B42V6 forcing directory:      ./input/FORCING/3B42V6/
```

```
TRMM 3B42V6 domain x-dimension size: 1440
```

```
TRMM 3B42V6 domain y-dimension size: 400
```

9.11.10 CMORPH precipitation

CMORPH forcing directory: specifies the location of the CMORPH forcing files.

CMORPH domain x-dimension size: specifies the number of columns of the native domain parameters of the CMORPH forcing data. The map projection is specified in the driver modules defined for the CMORPH routines.

CMORPH domain y-dimension size: specifies the number of rows of the native domain parameters of the CMORPH forcing data. The map projection is specified in the driver modules defined for the CMORPH routines.

```
CMORPH forcing directory:      ./input/FORCING/CMORPH/
```

```
CMORPH domain x-dimension size: 4989
```

```
CMORPH domain y-dimension size: 1649
```

9.11.11 Stage II precipitation

STAGE2 forcing directory: specifies the location of the STAGE2 forcing files.

```
STAGE2 forcing directory:          ./input/FORCING/STII
```

9.11.12 Stage IV precipitation

STAGE4 forcing directory: specifies the location of the STAGE4 forcing files.

```
STAGE4 forcing directory:          ./input/FORCING/STIV
```

9.11.13 D2PCPCAR

D2PCPCAR **forcing directory**: specifies the location of the D2PCPCAR forcing files.

D2PCPCAR **forcing directory**:

9.11.14 D2PCPOKL

D2PCPOKL **forcing directory**: specifies the location of the D2PCPOKL forcing files.

D2PCPOKL **forcing directory**:

9.11.15 NARR

NARR **forcing directory**: specifies the location of the NARR forcing files.

NARR **domain x-dimension size**: specifies the number of columns of the native domain parameters of the NARR forcing data.

NARR **domain y-dimension size**: specifies the number of rows of the native domain parameters of the NARR forcing data.

NARR **domain y-dimension size**: specifies the number of rows of the native domain parameters of the NARR forcing data.

NARR **domain z-dimension size**: specifies the number of atmospheric profiles in the NARR forcing data.

NARR forcing directory :	./input/Code/NARR/
NARR domain x-dimension size :	768
NARR domain y-dimension size :	386
NARR domain z-dimension size :	30

9.11.16 GDAS3D

GDAS3D forcing directory: specifies the location of the GDAS3D forcing files.
GDAS3D domain x-dimension size: specifies the number of columns of the native domain parameters of the GDAS3D forcing data.

GDAS3D domain y-dimension size: specifies the number of rows of the native domain parameters of the GDAS3D forcing data.

GDAS3D domain y-dimension size: specifies the number of rows of the native domain parameters of the GDAS3D forcing data.

GDAS3D domain z-dimension size: specifies the number of atmospheric profiles in the GDAS3D forcing data.

GDAS3D forcing directory:	./input/FORCING/GDAS3D/
GDAS3D domain x-dimension size:	768
GDAS3D domain y-dimension size:	386
GDAS3D domain z-dimension size:	64

9.11.17 ARMS

ARMS forcing file: specifies the location of the ARMS station met forcing file

ARMS precip forcing file: specifies the location of the ARMS precip station forcing file

ARMS station metadata file: specifies the location of the ARMS station metadata file

ARMS forcing file: .../WG_domain/1990_forc.txt

ARMS precip forcing file: .../WG_domain/WGbrk_19900723-19900815.out

ARMS station metadata file: .../WG_domain/WGbrk_stnlocs_1990.dat

9.12 Land surface models

9.12.1 Forcing only – Template

TEMPLATE model output interval: defines the output interval for the template LSM, in seconds. The template LSM is not a model; rather, it is a placeholder for a model. It demonstrates the hooks that are needed to add a land surface model into LIS. This “LSM” is also used to run LIS with the purpose of only processing and writing forcing data.

TEMPLATE model output interval: 3600 #in seconds

9.12.2 NCEP’s Noah

NOAH model output interval: defines the output interval for Noah, in seconds. The typical value used in the LIS runs is 3 hours (=10800)

NOAH restart output interval: defines the restart writing interval for Noah, in seconds. The typical value used in the LIS runs is 24 hours (=86400).

NOAH restart file: specifies the Noah active restart file.

NOAH slope file: specifies the Noah static slope file.

NOAH vegetation parameter table: specifies the Noah static vegetation parameter table file.

NOAH soil parameter table: specifies the Noah soil parameter file.

NOAH use PTF for mapping soil properties: specifies if pedotransfer functions are to be used for mapping soil properties (0-do not use, 1-use)

NOAH bottom temperature climatology interval: specifies in months, the climatology interval of the TBOT files. 0 indicates that the files are static.

NOAH number of vegetation parameters: specifies the number of static vegetation parameters specified for each veg type.

NOAH soils scheme: specifies the soil mapping scheme used. Acceptable values are:

Value	Description
1	Zobler
2	STATSGO

NOAH number of soil classes: specifies the number soil classes in the above mapping scheme Acceptable values are:

Value	Description
9	Zobler
19	STATSGO

NOAH number of soil layers: specifies the number of soil layers. The typical value used in Noah is 4.

NOAH observation height: specifies the height in meters of meteorological observations. The typical value used in Noah is 6 meters.

NOAH initial soil moisture: specifies the initial volumetric soil moisture used in the cold start runs. (units $\frac{m^3}{m^3}$)

NOAH initial soil temperature: specifies the initial skin temperature in Kelvin used in the cold start runs.

NOAH use forcing data observation height: specifies whether to use observation height from the forcing dataset.

Acceptable values are:

Value	Description
0	Do not use observation height from forcing
1	Use observation height from forcing

NOAH use forcing data aerodynamic conductance: specifies whether to use aerodynamic conductance field from the forcing dataset.

Acceptable values are:

Value	Description
0	Do not use aerodynamic conductance from forcing data
1	Use aerodynamic conductance from forcing dataset

```
NOAH model output interval:          10800
NOAH restart output interval:        86400
NOAH restart file:
NOAH slope file:
NOAH vegetation parameter table:    ./input/noah_parms/noah.vegparms.txt
NOAH soil parameter table:           ./input/noah_parms/noah.soilparms.txt
NOAH general parameter table:       ./input/noah_parms/GENPARM.UNIF.TBL
NOAH bottom temperature climatology interval: 0
NOAH number of vegetation parameters: 7
NOAH soils scheme:                  1
NOAH number of soil classes:         9
NOAH number of soil layers:          4
NOAH observation height:            20      #meters
NOAH initial soil moisture:          0.20    #volumetric soil moisture (m3 m-3)
NOAH initial soil temperature:       290.0   # Kelvin
NOAH use forcing data observation height: 0
NOAH use forcing data aerodynamic conductance: 0
```

9.12.3 CLM 2.0

CLM model output interval: defines the output interval for CLM, in seconds. The typical value used in the LIS runs is 3 hours (=10800)

CLM restart output interval: defines the restart writing interval for CLM, in seconds. The typical value used in the LIS runs is 24 hours (=86400).

CLM restart file: specifies the CLM active restart file. **CLM vegetation parameter file:** specifies vegetation type parameters look-up table.

CLM canopy height table: specifies the canopy top and bottom heights (for each vegetation type) look-up table.

CLM initial soil moisture: specifies the initial volumetric soil moisture wetness used in the cold start runs.

CLM initial soil temperature: specifies the initial soil temperature in Kelvin used in the cold start runs.

CLM initial snow mass: specifies the initial snow mass used in the cold start runs.

CLM model output interval:	10800
CLM restart output interval:	86400
CLM restart file:	
CLM vegetation parameter table:	./input/clm_parms/umdvegparam.txt
CLM canopy height table:	./input/clm_parms/clm2_ptcanhts.txt
CLM initial soil moisture:	0.45
CLM initial soil temperature:	290.0
CLM initial snow mass:	0.0

9.12.4 VIC

CLM model output interval: defines the output interval for CLM, in seconds. The typical value used in the LIS runs is 3 hours (=10800)

VIC restart output interval: defines the restart writing interval for VIC, in seconds. The typical value used in the LIS runs is 24 hours (=86400).

VIC number of soil layers: specifies the number of soil layers. The typical value used in VIC is 4.

VIC number of soil thermal nodes: specifies the number of soil thermal nodes. The typical value used in VIC is 4.

VIC number of snow bands: specifies the number of snow bands. The typical value used in VIC is 4.

VIC number of rootzones: specifies the number of rootzones. The typical value used in VIC is 4.

VIC full energy balance mode: specifies whether or not VIC will run in full energy balance mode.

Acceptable values are:

Value	Description
0	do not run in full energy mode
1	run in full energy mode

VIC frozen soil mode: specifies whether or not VIC will run in frozen soil mode. Acceptable values are:

Value	Description
0	do not run in frozen soil mode
1	run in frozen soil mode

VIC soil parameter table specifies the soil parameters look-up table.

VIC vegetation parameter table specifies the vegetation parameters look-up table.

VIC restart file: specifies the VIC active restart file. **VIC Ds map:** specifies the Ds map.

VIC DsMax map: specifies the DsMax map.

VIC Ws map: specifies the Ws map.

VIC infiltration capacity map: specifies the infiltration capacity map.

VIC depth1 map: specifies the depth1 map.

VIC depth2 map: specifies the depth2 map.

VIC depth3 map: specifies the depth3 map.

VIC initial surf temp: specifies the initial surface temperature.

VIC model output interval:

VIC restart output interval:

VIC number of soil layers:

VIC number of soil thermal nodes:

VIC number of snow bands:

VIC number of rootzones:

VIC full energy balance mode:

VIC frozen soil mode:

VIC soil parameter table:

VIC vegetation parameter table:

VIC restart file:

VIC Ds map:

VIC DsMax map:

VIC Ws map:

VIC infiltration capacity map:

VIC depth1 map:
VIC depth2 map:
VIC depth3 map:
VIC initial surf temp:

9.12.5 Mosaic

Mosaic model output interval: defines the output interval for Mosaic, in seconds. The typical value used in the LIS runs is 3 hours (=10800)

Mosaic restart output interval: defines the restart writing interval for Mosaic, in seconds. The typical value used in the LIS runs is 24 hours (=86400).

Mosaic restart file: specifies the Mosaic active restart file.

Mosaic vegetation parameter table: specifies the vegetation parameters look-up table.

Mosaic monthly vegetation parameter table: specifies the monthly vegetation parameters look-up table.

Mosaic soil parameter table: specifies the soil parameters look-up table.

Mosaic number of soil classes: specifies the number of soil classes. Acceptable values are:

Value	Description
11	FAO

Mosaic initial soil moisture: specifies the initial soil moisture.

Mosaic initial soil temperature: specifies the initial soil temperature in Kelvin.

Mosaic Depth of Layer 1 (m): specifies the depth in meters of layer 1.

Mosaic Depth of Layer 2 (m): specifies the depth in meters of layer 2.

Mosaic Depth of Layer 3 (m): specifies the depth in meters of layer 3.

Mosaic use forcing data observation height: specifies whether to use observation height from the forcing dataset.

Acceptable values are:

Value	Description
0	Do not use observation height from forcing
1	Use observation height from forcing

Mosaic use forcing data aerodynamic conductance: specifies whether to use aerodynamic conductance field from the forcing dataset.

Acceptable values are:

Value	Description
0	Do not use aerodynamic conductance from forcing data
1	Use aerodynamic conductance from forcing dataset

Mosaic model output interval: 10800
Mosaic restart output interval: 86400
Mosaic restart file:
Mosaic vegetation parameter table: ./input/mos_parms/mosaic_vegparms_umd.txt
Mosaic monthly vegetation parameter table: ./input/mos_parms/mosaic_monthlyvegparms_umd.txt
Mosaic soil parameter table: ./input/mos_parms/mosaic_soilparms_fao.txt
Mosaic number of soil classes: 11
Mosaic initial soil moisture: 0.3
Mosaic initial soil temperature: 290
Mosaic Depth of Layer 1 (m): 0.02
Mosaic Depth of Layer 2 (m): 1.48
Mosaic Depth of Layer 3 (m): 2.00
Mosaic use forcing data observation height: 0
Mosaic use forcing data aerodynamic conductance: 0

9.12.6 Hyssib

HYSSIB model output interval: defines the output interval for HYSSIB, in seconds. The typical value used in the LIS runs is 3 hours (=10800)

HYSSIB restart output interval: defines the restart writing interval for HYSSIB, in seconds. The typical value used in the LIS runs is 24 hours (=86400).

HYSSIB restart file: specifies the HYSSIB active restart file. **HYSSIB vegetation parameter table:** specifies the HYSSIB static vegetation parameter table file.

HYSSIB albedo parameter table: specifies the HYSSIB static albedo parameter table file.

HYSSIB topography stand dev file: specifies the HYSSIB topography standard deviation file.

HYSSIB bottom temperature climatology interval: specifies the interval of the HYSSIB bottom temperature climatology.

HYSSIB number of vegetation parameters: specifies the number of vegetation parameters.

HYSSIB number of vegetation parameters: specifies the number of monthly vegetation parameters.

HYSSIB initial soil moisture: specifies the initial soil moisture.

HYSSIB initial soil temperature: specifies the initial soil temperature in Kelvin.

```
HYSSIB model output interval: 10800
HYSSIB restart output interval: 86400
HYSSIB restart file:
HYSSIB vegetation parameter table: ./input/hyssib_parms/hyssib_vegparms.bin
HYSSIB albedo parameter table: ./input/hyssib_parms/hyssib_albedo.bin
HYSSIB topography stand dev file: ./input/UMD-25KM/topo_std.1gd4r
HYSSIB bottom temperature climatology interval: 0
HYSSIB number of vegetation parameters: 20
HYSSIB number of monthly veg parameters: 11
HYSSIB initial soil moisture: 0.30
HYSSIB initial soil temperature: 290.0
```

9.12.7 SiB2

SiB2 model output interval: defines the output interval for SiB2, in seconds. The typical value used in the LIS runs is 3 hours (=10800)

SiB2 restart output interval: defines the restart writing interval for SiB2, in seconds. The typical value used in the LIS runs is 24 hours (=86400).

SiB2 restart file: specifies the SiB2 active restart file.

SiB2 albedo and radiation parameter file: specifies the SiB2 albedo and radiation parameter file.

SiB2 monthly vegetation parameter file: specifies the monthly vegetation parameter file.

SiB2 static vegetation parameter file: specifies the static vegetation parameter file.

SiB2 initial soil moisture: specifies the initial volumetric soil moisture. (units $\frac{m^3}{m^3}$)

SiB2 initial soil temperature: specifies the initial soil temperature in Kelvin.

```
SiB2 model output interval: 10800
SiB2 restart output interval: 86400
SiB2 restart file:
SiB2 albedo and radiation parameter file: ./input/sib2_parms/ssibalb
SiB2 monthly vegetation parameter file: ./input/UMD-100KM/veg_month_1.0.1gd4r
SiB2 static vegetation parameter file: ./input/UMD-100KM/veg_const_1.0.1gd4r
SiB2 initial soil moisture: 0.20
SiB2 initial soil temperature: 290.0
```

9.12.8 Catchment

Catchment based tile map

tile coord file: specifies the tile coordinate file. This file maps the Catchment-based tiles to their overlying atmospheric grid.

tile veg file: specifies the tile vegetation fraction file. This file contains the fractions of the various vegetation types for Catchment-based tile.

Catchment model output interval: defines the output interval for Catchment, in seconds. The typical value used in the LIS runs is 3 hours (=10800)

Catchment restart output interval: defines the restart writing interval for Catchment, in seconds. The typical value used in the LIS runs is 24 hours (=86400).

Catchment restart file: specifies the Catchment active restart file.

Catchment roughness length table: specifies the Catchment roughness length table.

Catchment veg height table: specifies the Catchment vegetation height table.

Catchment parameter directory: specifies the directory containing the various Catchment parameter data. **Catchment MODIS directory:** specifies the directory containing the MODIS-based data.

ONLY USED FOR CATCHMENT-BASED DOMAIN TYPE. **Catchment veg class file:** specifies the vegetation classification file.

ONLY USED FOR CATCHMENT-BASED DOMAIN TYPE. **Catchment soil param file:** specifies the soil parameter file.

ONLY USED FOR CATCHMENT-BASED DOMAIN TYPE. **Catchment surf layer ts file:** specifies the surface layer ts file.

ONLY USED FOR CATCHMENT-BASED DOMAIN TYPE. **Catchment topo ar file:** specifies the topography ar file.

ONLY USED FOR CATCHMENT-BASED DOMAIN TYPE. **Catchment topo bf file:** specifies the topography bf file.

ONLY USED FOR CATCHMENT-BASED DOMAIN TYPE. **Catchment topo ts file:** specifies the topography ts file.

ONLY USED FOR CATCHMENT-BASED DOMAIN TYPE. **Catchment initial soil moisture:** specifies the initial volumetric soil moisture. (units $\frac{m^3}{m^3}$)

Catchment initial soil temperature: specifies the initial soil temperature in Kelvin.

```
tile coord file:      ./input/cat_parms/PE_360x180_DE_288x270_DE_NO_TINY.rst
tile veg file:       ./input/cat_parms/mosaic_veg_typs_fracs
Catchment model output interval:    10800
Catchment restart output interval:  86400
Catchment restart file:
```

```
Catchment roughness length table:  
Catchment veg height table:  
Catchment parameter directory:      ./input/cat_parms/  
Catchment MODIS directory:  
Catchment veg class file:  
Catchment soil param file:  
Catchment surf layer ts file:  
Catchment topo ar file:  
Catchment topo bf file:  
Catchment topo ts file:  
Catchment initial soil moisture:    0.30  
Catchment initial soil temperature: 290.0
```

9.13 Model output configuration

The output start time is used to define when to begin writing model output. Any value not defined will default to the corresponding LIS start time. The output start time does not affect restart writing. Restart files are written according to the LIS start time and the model restart output interval value.

The output start time is specified in the following format:

Variable	Value	Description
Output start year:	integer 2001 – present	specifying output start year
Output start month:	integer 1 – 12	specifying output start month
Output start day:	integer 1 – 31	specifying output start day
Output start hour:	integer 0 – 23	specifying output start hour
Output start minute:	integer 0 – 59	specifying output start minute
Output start second:	integer 0 – 59	specifying output start second

```
Output start year:  
Output start month:  
Output start day:  
Output start hour:  
Output start minutes:  
Output start seconds:
```

Model output attributes file: specifies the attributes to be used for a customizable model output. Please refer to the sample MODEL_OUTPUT_LIST.TBL file for the complete specification.

Model output attributes file: './MODEL_OUTPUT_LIST.TBL'

DRAFT

10 Specification of Input Forcing Variables

This section defines the specification of the input forcing variables for LIS. This file is specified in a space delimited column format. Each row consists of the following entries:

Short Name short name of the forcing variable.

Use option determines whether to include this the variable for use within LIS. Acceptable values are:

Value	Description
0	do not include the variable
1	include the variable

Number of vertical levels The number of vertical levels corresponding to the variable.

Units specified unit of the variable.

Note that this is a full list of input forcing variables. Not all models use all these variables.

```
#short name select vlevels units
Tair:      1  1   K      # Near Surface Air Temperature
Qair:      1  1   kg/kg  # Near Surface Specific Humidity
SWdown:    1  1   W/m2   # Incident Shortwave Radiation
SWdirect:  0  1   W/m2   # Incident Shortwave Radiation
SWdiffuse: 0  1   W/m2   # Incident Shortwave Radiation
LWdown:    1  1   W/m2   # Incident Longwave Radiation
Wind_E:    1  1   W/m2   # Eastward Wind
Wind_N:    1  1   m/s    # Northward Wind
Psurf:     1  1   Pa     # Surface Pressure
Rainf:     1  1   kg/m2s # Rainfall Rate
Snowf:     0  1   kg/m2s # Snowfall Rate
CRainf:    1  1   kg/m2s # Convective Rainfall Rate
Forc_Hgt:  0  1   m      # Height of Forcing Variables
Ch:        0  1   -       # Surface Exchange Coefficient for Heat
Cm:        0  1   -       # Surface Exchange Coefficient for Momentum
Q2sat:    0  1   -       # Saturated Mixing Ratio
Emiss:    0  1   -       # Surface Emissivity
Cosz:     0  1   -       # Cosine of Zenith Angle
Albedo:   0  1   -       # Surface Albedo
```

```
LPressure: 0 1 Pa      # Level pressure  
O3:        0 1 -       # Ozone concentration
```

DRAFT

11 Specification of Data Assimilation and Perturbation Attribute Files

This section defines the specification of the attribute files related to data assimilation and perturbation algorithms.

There are three sets of files to be defined for specifying perturbation attributes, for:

1. Forcing variables
2. Model prognostic state variables
3. Observations

In addition, the use of dynamic bias estimation algorithms require the specification of a bias options file.

11.1 Forcing perturbations specification

The *forcing_attribs.txt* file must be specified in the following format, where the full name of the forcing variable (include the vertical level) is specified first, followed by a line that lists the maximum and minimum allowable values for that variable.

```
#varmin varmax
Incident Shortwave Radiation Level 001
0. 1300.
Incident Longwave Radiation Level 001
-50. 800.
Near Surface Air Temperature Level 001
200 350
```

The *forcing_pertattribs.txt* file must be specified in the following format, where the full name of the forcing variable (include the vertical level) is specified first. This is followed by a line that lists the following specifications (in a space delimited column format) for that variable.

Perturbation Type

Acceptable values are:

Value	Description
0	Additive perturbations
1	Multiplicative perturbations

Standard Deviation Standard deviation of perturbations

Standard Normal Max Maximum allowed normalized perturbation relative to $N(0,1)$

zeromean Ensure zero mean across ensemble or not (1-ensure, 0-do not ensure)

tcorr temporal correlation length in seconds

xcorr correlation length along latitudes (deg)

ycorr correlation length along longitudes (deg)

ccorr cross-correlations between different kinds of perturbations

A sample specification is shown below:

```
#ptype    std    std_max   zeromean  tcorr  xcorr  ycorr  ccorr
Incident Shortwave Radiation Level 001
1        0.30    2.5 1    86400   0      0      1.0    -0.5   0.0
Incident Longwave Radiation Level 001
0        15.0     2.5 1    86400   0      0      -0.5    1.0    0.0
Near Surface Air Temperature Level 001
0        1.0      2.5 1    86400   0      0      0.0    0.0    1.0
```

11.2 Model prognostic state perturbations specification

The *state_attribs.txt* file must be specified in the following format, where the full name of the state variable is specified first, followed by a line that lists the maximum and minimum allowable values for that variable.

A sample specification for Noah LSM's 4 layer soil moisture prognostics are shown below

```
#name  varmin  varmax
Soil Moisture Layer 1
  0.1 0.55
Soil Moisture Layer 2
  0.1 0.55
Soil Moisture Layer 3
  0.1 0.55
Soil Moisture Layer 4
  0.1 0.55
```

The *state_pertattribs.txt* file must be specified in the following format, where the full name of the state variable is specified first. This is followed by a line that lists the following specifications (in a space delimited column format) for that variable.

Perturbation Type

Acceptable values are:

Value Description

0 Additive perturbations

1 Multiplicative perturbations

Standard Deviation Standard deviation of perturbations

Standard Normal Max Maximum allowed normalized perturbation relative to $N(0,1)$

zeromean Ensure zero mean across ensemble or not (1-ensure, 0-do not ensure)

tcorr temporal correlation length in seconds

xcorr correlation length along latitudes (deg)

ycorr correlation length along longitudes (deg)

ccorr cross-correlations between different kinds of perturbations

A sample specification is shown below:

```
#perttype std std_max zeromean tcorr xcorr ycorr ccorr
Soil Moisture Layer 1
 0 0.6E-3 2.5 1 43200 0 0 1.0 0.6 0.4 0.2
Soil Moisture Layer 2
 0 1.1E-4 2.5 1 43200 0 0 0.6 1.0 0.6 0.4
Soil Moisture Layer 3
 0 0.6E-4 2.5 1 43200 0 0 0.4 0.6 1.0 0.6
Soil Moisture Layer 4
 0 0.4E-4 2.5 1 43200 0 0 0.2 0.4 0.6 1.0
```

11.3 Observations perturbations specification

The *obs_attribs.txt* file must be specified in the following format, where the full name of the observation variable is specified first, followed by a line that lists the maximum and minimum allowable values for that variable.

A sample specification for ISCCP skin temperature observations are shown below

```
#error rate varmin varmax
ISCCP skin temperature
220 400
```

The *obs_pertattribs.txt* file must be specified in the following format, where the full name of the observation variable is specified first. This is followed by a line that lists the following specifications (in a space delimited column format) for that variable.

Perturbation Type

Acceptable values are:

Value Description

0 Additive perturbations

1 Multiplicative perturbations

Standard Deviation Standard deviation of perturbations

Standard Normal Max Maximum allowed normalized perturbation relative to $N(0,1)$

zeromean Ensure zero mean across ensemble or not (1-ensure, 0-do not ensure)

tcorr temporal correlation length in seconds

xcorr correlation length along latitudes (deg)

ycorr correlation length along longitudes (deg)

ccorr cross-correlations between different kinds of perturbations

A sample specification is shown below:

```
#perttype std std_max zeromean tcorr xcorr ycorr ccorr
ISCCP skin temperature
0 1.0 2.5 1 0 0 0 1
```

11.4 Bias options specifications

The *bias_atrrib.txt* file must be specified in the following format, where the full name of the observation variable is specified first. This is followed by a line that lists the following specifications (in a space delimited column format) for that variable.

nparam number of estimated bias parameters per model field

Acceptable values are:

Value Description

0 no bias correction

1 constant bias correction (without diurnal cycle)

3 diurnal sine/cosine bias correction

5 semi-diurnal sine/cosine bias correction

2 "time-of-day" bias correction with 2 separate bias estimates per day

4 "time-of-day" bias correction with 4 separate bias estimates per day

8 "time-of-day" bias correction with 8 separate bias estimates per day

tconst time scale relative to the temporal spacing of observations (dimensionless)

trelax temporal relaxation value (in seconds)

A sample file is shown below

```
#nparam tconst trelax
Soil Temperature
1.0 0.05 86400.0
```

12 Specification of Parameter Estimation Attribute Files

This section defines the specification of the attribute files related to parameter estimation algorithms.

12.1 Decision space specification for Genetic Algorithm

A specification of the decision space variables for use within the Genetic Algorithm must be specified in the following format.

Nparam Number of parameters being estimated

Names Names of parameters being estimated (one in each line)

Max Maximum allowable values of each parameters (space delimited, column format)

Min Minimum allowable values of each parameters (space delimited, column format)

Npossible Number of parameter possibilities

A sample specification is shown below:

```
#Nparam  
2  
#Names  
Sand Fraction  
Clay Fraction  
# Max  
1.0 1.0  
# Min  
0.0001 0.0001  
#npossible  
32768 32768
```

13 Model Output Specifications

This section defines the specification of the model output from LIS. This file is specified in a space delimited column format. Each row consists of the following entries:

Short Name ALMA compliant short name of the variable.

Use option determines whether to write the variable. Acceptable values are:

Value	Description
0	do not write the variable
1	write the variable

Units the desired unit of the output variable.

Time Average option determines how temporally process the variable. Acceptable values are:

Value	Description
0	Instantaneous output
1	Time averaged output
2	Instantaneous and Time averaged output
3	Accumulated output

Min/Max option determines whether to record minimum and maximum values for the variable. Acceptable values are:

Value	Description
0	Do no compute minimum and maximum values
1	Do compute minimum and maximum values

Number of vertical levels The number of vertical levels corresponding to the variable.

grib ID The grib ID to be used for the variable if output is written in grib1 format.

grib scale factor The grib scale factor to be used for the variable if output is written in grib1 format.

Note that this is a full list of output variables. Not all models support all these variables. You must check the source code to verify that the model you

want to run supports the variables that you want to write.

```
#short_name select? units timeavg? min/max? vert.levels grib_id grib_scalefactor longname
#Energy balance components
Swnet:      1 W/m2  1 0 1 111 10 # Net Shortwave Radiation (W/m2)
Lwnet:      1 W/m2  1 0 1 112 10 # Net Longwave Radiation (W/m2)
Qle:        1 W/m2  1 0 1 121 10 # Latent Heat Flux (W/m2)
Qh:         1 W/m2  1 0 1 122 10 # Sensible Heat Flux (W/m2)
Qg:         1 W/m2  1 0 1 155 10 # Ground Heat Flux (W/m2)
Qf:         0 W/m2  1 0 1 229 10 # Energy of fusion (W/m2)
Qv:         0 W/m2  1 0 1 134 10 # Energy of sublimation (W/m2)
Qa:         0 W/m2  1 0 1 136 10 # Advection Energy (W/m2)
Qtau:       0 W/m2  1 0 1 135 10 # Momentum flux (N/m2)
DelSurfHeat: 0 W/m2  1 0 1 137 10 # Change in surface heat storage (J/m2)
DelColdCont: 0 W/m2  1 0 1 138 10 # Change in snow cold content (J/m2)

#Water balance components
Snowf:      1 kg/m2s 1 0 1 161 10000 # Snowfall rate (kg/m2s)
Rainf:      1 kg/m2s 1 0 1 162 10000 # Rainfall rate (kg/m2s)
RainfConv:   1 kg/m2s 1 0 1 163 10000 # Convective Rainfall rate (kg/m2s)
TotalPrecip: 1 kg/m2s 1 0 1 164 10000 # Total Precipitation rate (kg/m2s)
Evap:        1 kg/m2s 1 0 1 57 10000 # Total Evapotranspiration (kg/m2s)
Qs:          1 kg/m2s 1 0 1 235 10000 # Surface runoff (kg/m2s)
Qrec:        0 kg/m2s 1 0 1 143 10000 # Recharge (kg/m2s)
Qsb:         1 kg/m2s 1 0 1 254 10000 # Subsurface runoff (kg/m2s)
Qsm:         0 kg/m2s 1 0 1 99 10000 # Snowmelt (kg/m2s)
Qfz:         0 kg/m2s 1 0 1 146 10000 # Refreezing of water in the snowpack (kg/m2s)
Qst:         0 kg/m2s 1 0 1 147 10000 # Snow throughfall (kg/m2s)
DelSoilMoist: 0 kg/m2 0 0 1 148 10000 # Change in soil moisture (kg/m2)
DelSWE:      0 kg/m2 0 0 1 149 1000 # Change in snow water equivalent (kg/m2)
DelSurfStor: 0 kg/m2 1 0 1 150 1000 # Change in surface water storage (kg/m2)
DelIntercept: 0 kg/m2 1 0 1 151 1000 # Change in interception storage (kg/m2)
RHMin:       0 - 0 0 1 51 10 # Minimum 2 meter relative humidity (-)

#Surface State Variables
SnowT:      0 K 1 0 1 152 10 # Snow surface temperature (K)
VegT:        0 K 1 0 1 153 10 # Vegetation canopy temperature (K)
BareSoilT:   0 K 1 0 1 154 10 # Temperature of bare soil (K)
AvgSurfT:   1 K 0 0 1 148 10 # Average surface temperature (K)
RadT:        0 K 1 0 1 156 10 # Surface Radiative Temperature (K)
Albedo:      1 - 0 0 1 84 100 # Surface Albedo (-)
SWE:         1 kg/m2 0 0 1 65 1000 # Snow Water Equivalent (kg/m2)
SnowDepth:   0 m 0 0 1 66 1000 # Snow Depth (m)
Snowcover:   0 m 0 0 1 66 100 # Snow Depth (m)
SWEVeg:      0 kg/m2 1 0 1 159 1000 # SWE intercepted by vegetation (kg/m2)
```

```

SurfStor:      0    kg/m2  1 0 1 160 1000 # Surface water storage (kg/m2)
SLiqFrac:      0    -      0 0 1 65 1000 # fraction of SWE in the liquid phase

#Subsurface State Variables
SoilMoist:     1    kg/m2  0 0 4 86 1000 # Average layer soil moisture (kg/m2)
SoilTemp:       1    K      0 0 4 85 1000 # Average layer soil temperature (K)
SmLiqFrac:     0    -      1 0 4 85 100 # Average layer fraction of liquid moisture (-)
SmFrozFrac:    0    -      1 0 4 85 100 # Average layer fraction of frozen moisture (-)
SoilWet:        0    -      0 0 1 85 100 # Total soil wetness (-)
RelSMC:         0    m3/m3 0 0 1 86 1000 # Relative soil moistutre
RootTemp:       0    K      0 0 1 85 1000 # Rootzone temperature (K)

#Evaporation components
PotEvap:        0    kg/m2s 1 0 1 166 1      # Potential Evapotranspiration (kg/m2s)
ECanop:          0    kg/m2s 1 0 1 200 1      # Interception evaporation (kg/m2s)
TVeg:            0    kg/m2s 1 0 1 210 1      # Vegetation transpiration (kg/m2s)
ESoil:           0    kg/m2s 1 0 1 199 1      # Bare soil evaporation (kg/m2s)
EWater:          0    kg/m2s 1 0 1 170 1      # Open water evaporation (kg/m2s)
RootMoist:       0    kg/m2  0 0 1 171 1      # Root zone soil moisture (kg/m2)
CanopInt:        0    kg/m2  0 0 1 223 1000 # Total canopy water storage (kg/m2)
EvapSnow:        0    kg/m2s 1 0 1 173 1000 # Snow evaporation (kg/m2s)
SubSnow:         0    kg/m2s 1 0 1 198 1000 # Snow sublimation (kg/m2s)
SubSurf:         0    kg/m2s 1 0 1 175 1000 # Sublimation of the snow free area (kg/m2s)
ACond:           0    m/s   1 0 1 179 100000 # Aerodynamic conductance
CCond:           0    m/s   1 0 1 179 1000000 # Canopy conductance

#Forcings
Wind_f:          1    m/s   0 0 1 177 10    # Near Surface Wind (m/s)
Rainf_f:          1    kg/m2s 0 0 1 162 1000 # Average rainfall rate
Snowf_f:          0    kg/m2s 0 0 1 161 1000 # Average snowfall rate
Tair_f:           1    K     0 0 1 11 10    # Near surface air temperature
Qair_f:           1    kg/kg 0 0 1 51 1000 # Near surface specific humidity
Psurf_f:          1    Pa    0 0 1 1 10    # Surface pressure
SWdown_f:         1    W/m2  0 0 1 204 10    # Surface incident shortwave radiation
LWdown_f:         1    W/m2  0 0 1 205 10    # Surface incident longwave radiation

#Parameters
Landmask:        0    -    0 0 1 81 1    # Land Mask (0 - Water, 1- Land)
Landcover:        0    -    0 0 1 186 1    # Land cover
Soiltype:         0    -    0 0 1 187 1    # soil type
SandFrac:         0    -    0 0 1 999 1    # sand fraction
ClayFrac:         0    -    0 0 1 999 1    # clay fraction
SiltFrac:         0    -    0 0 1 999 1    # silt fraction
Porosity:         0    -    3 0 1 999 1    # porosity
Soilcolor:        0    -    0 0 1 188 1    # soil color
Elevation:        0    m    0 0 1 189 10    # elevation

```

```
Slope:      0   -  0 0 1 999 10  # slope
LAI:       0   -  0 0 1 190 100 # LAI
SAI:       0   -  0 0 1 191 100 # SAI
Snfralbedo: 0   -  0 0 1 192 100 #
Mxsnalbedo: 0   -  0 0 1 192 100 #
Greenness:  0   -  0 0 1   87 100 #
Tempbot:   0   -  0 0 1 194 10  #
```

DRAFT

A Cylindrical Lat/Lon Domain Example

This section describes how to compute the values for the run domain and param domain sections on a cylindrical lat/lon projection.

First, we shall generate the values for the parameter data domain. LIS' parameter data is defined on a Latitude/Longitude grid, from -180 to 180 degrees longitude and from -60 to 90 degrees latitude.

For this example, consider running at $1/4$ deg resolution. The coordinates of the south-west and the north-east points are specified at the grid-cells' centers. Here the south-west grid-cell is given by the box $(-180, -60), (-179.750, -59.750)$. The center of this box is $(-179.875, -59.875)$.¹

```
param domain lower left lat: -59.875  
param domain lower left lon: -179.875
```

The north-east grid-cell is given by the box $(179.750, 89.750), (180, 90)$. Its center is $(179.875, 89.875)$.

```
param domain upper right lat: 89.875  
param domain upper right lon: 179.875
```

Setting the resolution (0.25 deg) gives

```
param domain resolution dx: 0.25  
param domain resolution dy: 0.25
```

And this completely defines the parameter data domain.

Next, we shall generate the values for the running domain.

If you wish to run over the whole domain defined by the parameter data domain then you simply set the values defined in the parameter domain section in the run domain section. This gives:

```
run domain lower left lat: -59.875  
run domain lower left lon: -179.875  
run domain upper right lat: 89.875  
run domain upper right lon: 179.875  
run domain resolution dx: 0.25  
run domain resolution dy: 0.25
```

Now say you wish to run only over the region given by $(-97.6, 27.9), (-92.9, 31.9)$. Since the running domain is a sub-set of the parameter domain, it is also a Latitude/Longitude domain at $1/4$ deg. resolution. Thus,

```
run domain resolution dx: 0.25  
run domain resolution dy: 0.25
```

Now, since the running domain must fit onto the parameter domain, the desired running region must be expanded from $(-97.6, 27.9), (-92.9, 31.9)$ to $(-97.75, 27.75), (-92.75, 32.0)$. The south-west grid-cell for the running domain is the box $(-97.75, 27.75), (-97.5, 28.0)$. Its center is $(-97.625, 27.875)$; giving

¹Note, these coordinates are ordered (longitude, latitude).

```
run domain lower left lat: 27.875  
run domain lower left lon: -97.625
```

The north-east grid-cell for the running domain is the box $(-93, 31.75), (-92.75, 32.0)$. Its center is $(-92.875, 31.875)$; giving

```
run domain upper right lat: 31.875  
run domain upper right lon: -92.875
```

This completely defines the running domain.

Note, the LIS project has defined 5 km resolution to be 0.05 deg. and 1 km resolution to be 0.01 deg. If you wish to run at 5 km or 1 km resolution, redo the above example to compute the appropriate grid-cell values.

See Figure 1 for an illustration of adjusting the running grid. See Figures 2 and 3 for an illustration of the south-west and north-east grid-cells.

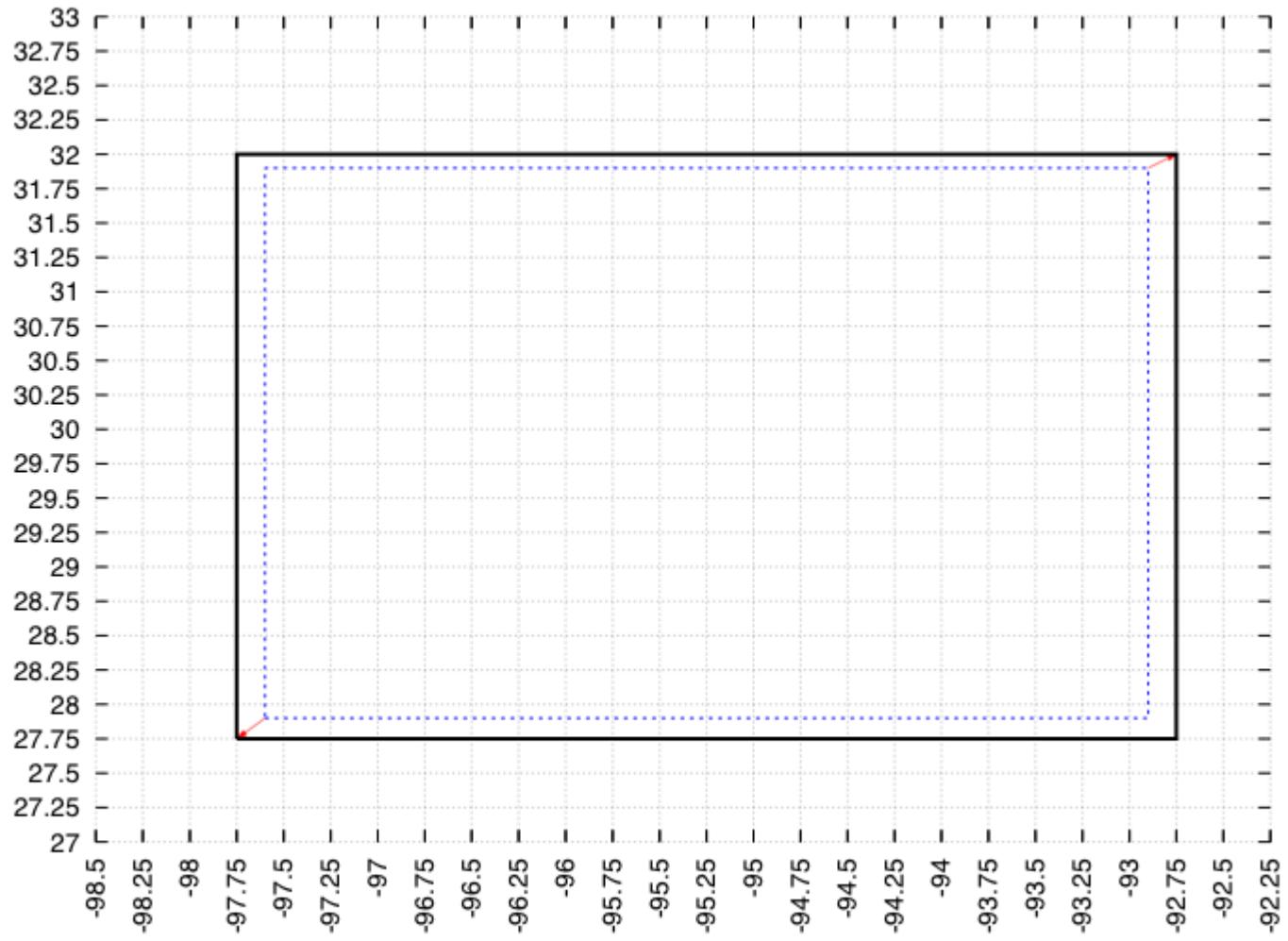


Figure 1: Illustration showing how to fit the desired running grid onto the actual grid

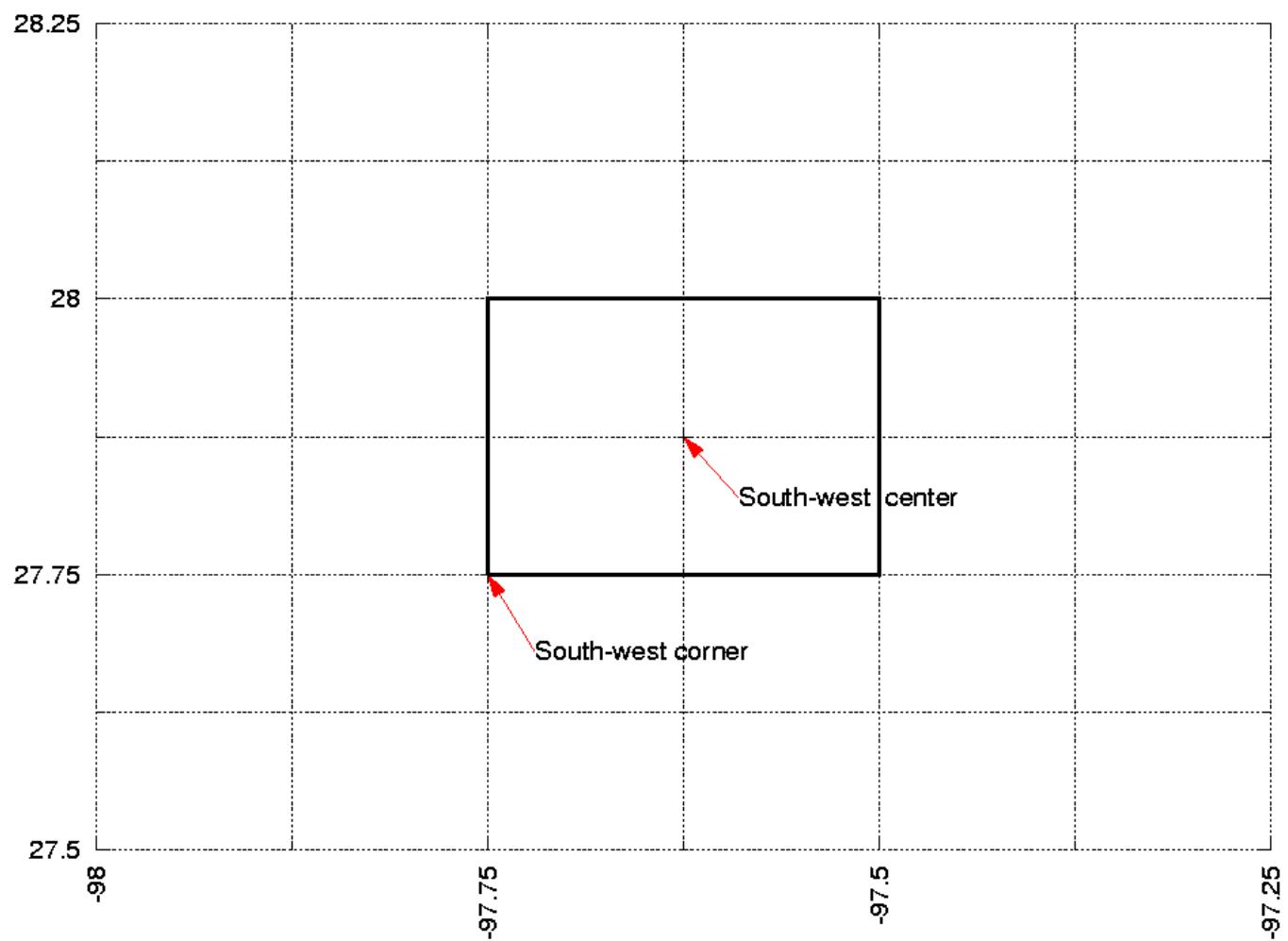


Figure 2: Illustration showing the south-west grid-cell corresponding to the example in Section A

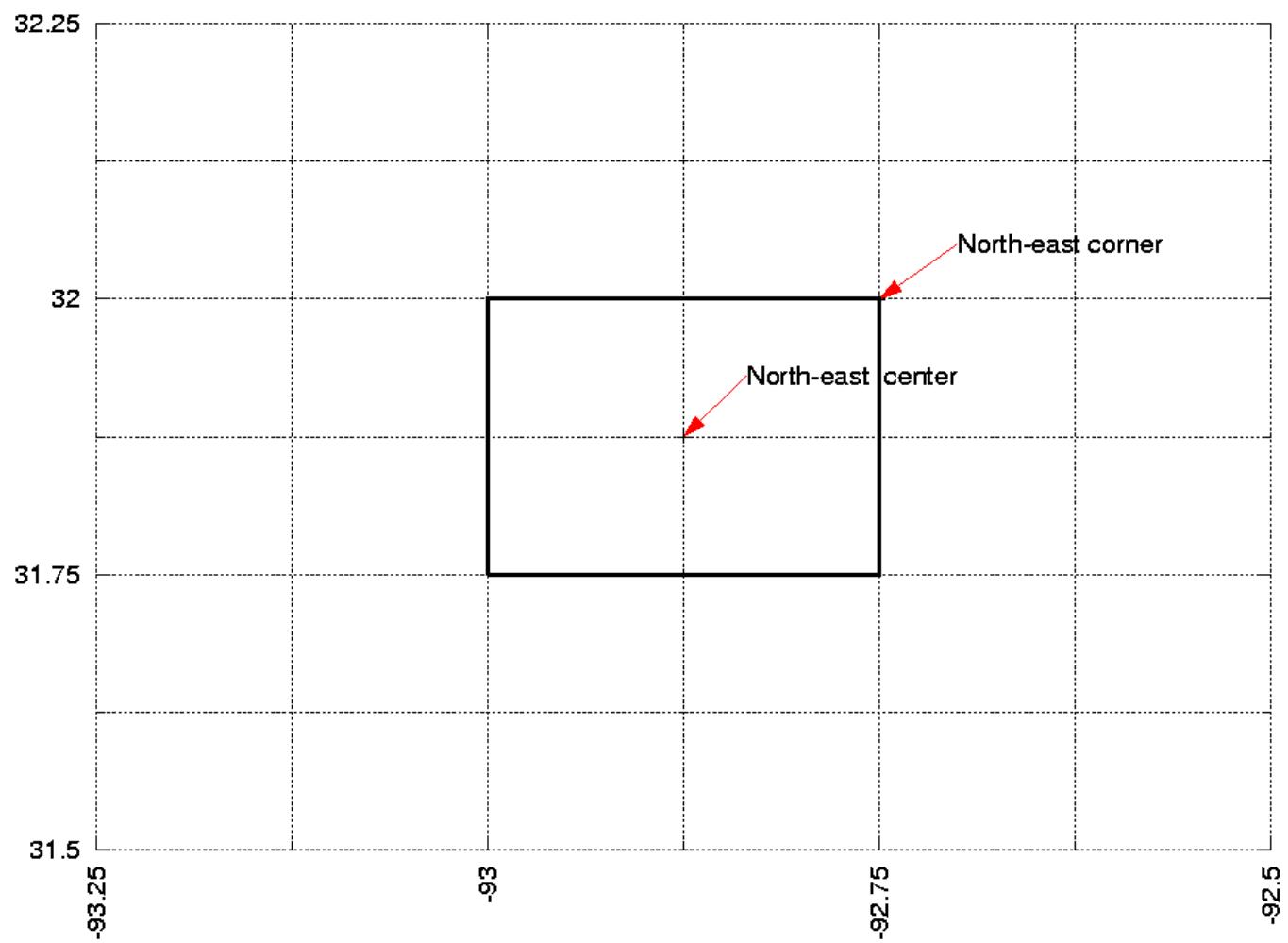


Figure 3: Illustration showing the north-east grid-cell corresponding to the example in Section A

B Polar Stereographic Domain Example

This section describes how to compute the values for the run domain and param domain sections on a polar stereographic projection.

STUB!

DRAFT

C Gaussian Domain Example

This section describes how to compute the values for the run domain and param domain sections on a Gaussian projection.

First, we shall generate the values for the parameter data domain. LIS' Gaussian parameter data is defined from -180 to 180 degrees longitude and from -90 to 90 degrees latitude. Note that the first longitude point is at 0 .

The parameter domain must be specified as follows:

```
param domain first grid point lat:      -89.27665
param domain first grid point lon:      0.0
param domain last grid point lat:       89.27665
param domain last grid point lon:      -0.9375
param domain resolution dlon:          0.9375
param domain number of lat circles:    95
```

Next, we shall generate the values for the running domain.

If you wish to run over the whole domain defined by the parameter data domain then you simply set the values defined in the parameter domain section in the run domain section. This gives:

```
run domain first grid point lat:      -89.27665
run domain first grid point lon:      0.0
run domain last grid point lat:       89.27665
run domain last grid point lon:      -0.9375
run domain resolution dlon:          0.9375
run domain number of lat circles:    95
```

If you wish to run over a sub-domain, then you must choose longitude and latitude values that correspond to the T126 Gaussian projection. Tables of acceptable longitude and latitude values are found below.

Now say you wish to run only over the region given by $(-97.6, 27.9), (-92.9, 31.9)$. Since the running domain must fit on the T126 Gaussian grid, the running domain must be expanded to $(-98.4375, 27.87391), (-91.875, 32.59830)$. Thus the running domain specification is:

```
run domain first grid point lat:      27.87391
run domain first grid point lon:      -98.4375
run domain last grid point lat:       32.59830
run domain last grid point lon:      -91.875
run domain resolution dlon:          0.9375
run domain number of lat circles:    95
```

Table 1: Acceptable longitude values

0.000000	0.937500	1.875000	2.812500	3.750000
4.687500	5.625000	6.562500	7.500000	8.437500
9.375000	10.312500	11.250000	12.187500	13.125000
14.062500	15.000000	15.937500	16.875000	17.812500
18.750000	19.687500	20.625000	21.562500	22.500000
23.437500	24.375000	25.312500	26.250000	27.187500
28.125000	29.062500	30.000000	30.937500	31.875000
32.812500	33.750000	34.687500	35.625000	36.562500
37.500000	38.437500	39.375000	40.312500	41.250000
42.187500	43.125000	44.062500	45.000000	45.937500
46.875000	47.812500	48.750000	49.687500	50.625000
51.562500	52.500000	53.437500	54.375000	55.312500
56.250000	57.187500	58.125000	59.062500	60.000000
60.937500	61.875000	62.812500	63.750000	64.687500
65.625000	66.562500	67.500000	68.437500	69.375000
70.312500	71.250000	72.187500	73.125000	74.062500
75.000000	75.937500	76.875000	77.812500	78.750000
79.687500	80.625000	81.562500	82.500000	83.437500
84.375000	85.312500	86.250000	87.187500	88.125000
89.062500	90.000000	90.937500	91.875000	92.812500
93.750000	94.687500	95.625000	96.562500	97.500000
98.437500	99.375000	100.312500	101.250000	102.187500
103.125000	104.062500	105.000000	105.937500	106.875000
107.812500	108.750000	109.687500	110.625000	111.562500
112.500000	113.437500	114.375000	115.312500	116.250000
117.187500	118.125000	119.062500	120.000000	120.937500
121.875000	122.812500	123.750000	124.687500	125.625000
126.562500	127.500000	128.437500	129.375000	130.312500
131.250000	132.187500	133.125000	134.062500	135.000000
135.937500	136.875000	137.812500	138.750000	139.687500
140.625000	141.562500	142.500000	143.437500	144.375000
145.312500	146.250000	147.187500	148.125000	149.062500
150.000000	150.937500	151.875000	152.812500	153.750000
154.687500	155.625000	156.562500	157.500000	158.437500
159.375000	160.312500	161.250000	162.187500	163.125000
164.062500	165.000000	165.937500	166.875000	167.812500
168.750000	169.687500	170.625000	171.562500	172.500000
173.437500	174.375000	175.312500	176.250000	177.187500
178.125000	179.062500	180.000000	-179.062500	-178.125000

-177.187500	-176.250000	-175.312500	-174.375000	-173.437500
-172.500000	-171.562500	-170.625000	-169.687500	-168.750000
-167.812500	-166.875000	-165.937500	-165.000000	-164.062500
-163.125000	-162.187500	-161.250000	-160.312500	-159.375000
-158.437500	-157.500000	-156.562500	-155.625000	-154.687500
-153.750000	-152.812500	-151.875000	-150.937500	-150.000000
-149.062500	-148.125000	-147.187500	-146.250000	-145.312500
-144.375000	-143.437500	-142.500000	-141.562500	-140.625000
-139.687500	-138.750000	-137.812500	-136.875000	-135.937500
-135.000000	-134.062500	-133.125000	-132.187500	-131.250000
-130.312500	-129.375000	-128.437500	-127.500000	-126.562500
-125.625000	-124.687500	-123.750000	-122.812500	-121.875000
-120.937500	-120.000000	-119.062500	-118.125000	-117.187500
-116.250000	-115.312500	-114.375000	-113.437500	-112.500000
-111.562500	-110.625000	-109.687500	-108.750000	-107.812500
-106.875000	-105.937500	-105.000000	-104.062500	-103.125000
-102.187500	-101.250000	-100.312500	-99.375000	-98.437500
-97.500000	-96.562500	-95.625000	-94.687500	-93.750000
-92.812500	-91.875000	-90.937500	-90.000000	-89.062500
-88.125000	-87.187500	-86.250000	-85.312500	-84.375000
-83.437500	-82.500000	-81.562500	-80.625000	-79.687500
-78.750000	-77.812500	-76.875000	-75.937500	-75.000000
-74.062500	-73.125000	-72.187500	-71.250000	-70.312500
-69.375000	-68.437500	-67.500000	-66.562500	-65.625000
-64.687500	-63.750000	-62.812500	-61.875000	-60.937500
-60.000000	-59.062500	-58.125000	-57.187500	-56.250000
-55.312500	-54.375000	-53.437500	-52.500000	-51.562500
-50.625000	-49.687500	-48.750000	-47.812500	-46.875000
-45.937500	-45.000000	-44.062500	-43.125000	-42.187500
-41.250000	-40.312500	-39.375000	-38.437500	-37.500000
-36.562500	-35.625000	-34.687500	-33.750000	-32.812500
-31.875000	-30.937500	-30.000000	-29.062500	-28.125000
-27.187500	-26.250000	-25.312500	-24.375000	-23.437500
-22.500000	-21.562500	-20.625000	-19.687500	-18.750000
-17.812500	-16.875000	-15.937500	-15.000000	-14.062500
-13.125000	-12.187500	-11.250000	-10.312500	-9.375000
-8.437500	-7.500000	-6.562500	-5.625000	-4.687500
-3.750000	-2.812500	-1.875000	-0.937500	

Table 2: Acceptable latitude values

-89.27665	-88.33975	-87.39729	-86.45353	-85.50930
-84.56487	-83.62028	-82.67562	-81.73093	-80.78618
-79.84142	-78.89662	-77.95183	-77.00701	-76.06219
-75.11736	-74.17252	-73.22769	-72.28285	-71.33799
-70.39314	-69.44830	-68.50343	-67.55857	-66.61371
-65.66885	-64.72399	-63.77912	-62.83426	-61.88939
-60.94452	-59.99965	-59.05478	-58.10991	-57.16505
-56.22018	-55.27531	-54.33043	-53.38556	-52.44069
-51.49581	-50.55094	-49.60606	-48.66119	-47.71632
-46.77144	-45.82657	-44.88169	-43.93681	-42.99194
-42.04707	-41.10219	-40.15731	-39.21244	-38.26756
-37.32268	-36.37781	-35.43293	-34.48805	-33.54317
-32.59830	-31.65342	-30.70854	-29.76366	-28.81879
-27.87391	-26.92903	-25.98415	-25.03928	-24.09440
-23.14952	-22.20464	-21.25977	-20.31489	-19.37001
-18.42513	-17.48025	-16.53537	-15.59050	-14.64562
-13.70074	-12.75586	-11.81098	-10.86610	-9.921225
-8.976346	-8.031467	-7.086589	-6.141711	-5.196832
-4.251954	-3.307075	-2.362196	-1.417318	-0.4724393
0.4724393	1.417318	2.362196	3.307075	4.251954
5.196832	6.141711	7.086589	8.031467	8.976346
9.921225	10.86610	11.81098	12.75586	13.70074
14.64562	15.59050	16.53537	17.48025	18.42513
19.37001	20.31489	21.25977	22.20464	23.14952
24.09440	25.03928	25.98415	26.92903	27.87391
28.81879	29.76366	30.70854	31.65342	32.59830
33.54317	34.48805	35.43293	36.37781	37.32268
38.26756	39.21244	40.15731	41.10219	42.04707
42.99194	43.93681	44.88169	45.82657	46.77144
47.71632	48.66119	49.60606	50.55094	51.49581
52.44069	53.38556	54.33043	55.27531	56.22018
57.16505	58.10991	59.05478	59.99965	60.94452
61.88939	62.83426	63.77912	64.72399	65.66885
66.61371	67.55857	68.50343	69.44830	70.39314
71.33799	72.28285	73.22769	74.17252	75.11736
76.06219	77.00701	77.95183	78.89662	79.84142
80.78618	81.73093	82.67562	83.62028	84.56487
85.50930	86.45353	87.39729	88.33975	89.27665

D Lambert Conformal Domain Example

This section describes how to compute the values for the run domain and param domain sections on a Lambert conformal projection.

STUB!

DRAFT

E Mercator Domain Example

This section describes how to compute the values for the run domain and param domain sections on a Mercator projection.

STUB!

DRAFT

F UTM Domain Example

This section describes how to compute the values for the run domain and param domain sections on a UTM projection.

DRAFT

G configure.lis

This is a sample *configure.lis* file used for compiling LIS on an IBM AIX system.

```
FC          = mpxlf90_r
FC77        = mpxlf_r
LD          = mpxlf90_r
CC          = mpcc_r
INC_NETCDF  = /usrx/local/netcdf/include
LIB_NETCDF  = /usrx/local/netcdf
LIB_MPI     = /usr/lpp/ppe.poe/lib
INC_MPI     = /usr/lpp/ppe.poe/include
LIB_ESMF    = /u/home/sujay/esmf_2_2_0rp1/lib/lib0/AIX.default.64.default
MOD_ESMF    = /u/home/sujay/esmf_2_2_0rp1/mod/mod0/AIX.default.64.default
CFLAGS      = -c -w -g -q64 -qcpluscmt -DSPMD
FFLAGS      = -I$(INC_MPI) -I$(MOD_ESMF) -c -g -qkeepparm \
              -qsuffix=f=F90:cpp=F90 -q64 -WF,-DAIX,-DSPMD
FFLAGS77    = -I$(INC_MPI) -I$(MOD_ESMF) -c -g -qkeepparm \
              -qsuffix=f=f:cpp=F90 -q64 -WF,-DAIX,-DSPMD
LDFLAGS     = -q64 -bmap:map -bloadmap:lm -lmass \
              ../lib/w3lib/libw3.a \
              ../lib/read_grib.aix/readgrib.a \
              ../lib/grib/griblib.a \
              -L$(LIB_ESMF) -lesmf -lnetcdf_stubs -lc_r
```

H READ GRIB - Information and Instructions

Thanks to Kristi Arsenault for putting this together

Caveat:

This is a package of subroutines to read GRIB-formatted data. It is still under continuous development. It won't be able to read every GRIB dataset you give it, but it will read a good many.

- Kevin W. Manning
NCAR/MMM
Summer 1998, and continuing

The main user interfaces are:

SUBROUTINE GRIBGET(NUNIT, IERR) - Read a single GRIB record from UNIX file-descriptor NUNIT into array GREC. No unpacking of any header or data values is performed.

SUBROUTINE GRIBREAD(NUNIT, DATA, NDATA, IERR) - Read a single GRIB record from UNIX file-descriptor NUNIT, and unpack all header and data values into the appropriate arrays.

SUBROUTINE GRIBHEADER(IERR) - Unpack the header of a GRIB record

SUBROUTINE GRIBDATA(DATARRAY, NDAT) - Unpack the data in a GRIB record into array DATARRAY

SUBROUTINE GRIBPRINT(ISEC) - Print the header information from GRIB section ISEC.

SUBROUTINE GET_SEC1(KSEC1) - Return the header information from Section 1.

SUBROUTINE GET_SEC2(KSEC2) - Return the header information from Section 2.

SUBROUTINE GET_GRIDINFO(IGINFO, GINFO) - Return the grid information of the previously-unpacked GRIB header.

C-ROUTINE – COPEN(UNIT, NUNIT, NAME, MODE, ERR, OFLAG) - Opens the GRIB file for reading later by the other routines.

H.1 SUBROUTINE GRIBGET (NUNIT, IERR)

- Read a single GRIB record from UNIX file-descriptor NUNIT into array GREC. No unpacking of any header or data values is performed.

- NOTE!: Intrinsic parameter, ied, identifies type of GRIB edition of GRIB file trying to open. Below are the codes (also see Table3):

- If IED == 1, then GRIB Edition 1 has the size of the whole GRIB record right up front.
- If IED == 0, then GRIB Edition 0 does not include the total size, so we have to sum up the sizes of the individual sections

- If IED > 1, then STOP, if not GRIB Edition 0 or 1

Input: NUNIT (integer): C unit number to read from. This should already be open.

Output: IERR (integer): Error flag, Non-zero means there was a problem with the read.

Side Effects: The array GREC is allocated, and filled with one GRIB record. The C unit pointer is moved to the end of the GRIB record just read.

H.2 SUBROUTINE GRIBREAD (NUNIT, DATA, NDATA, IERR)

- Read a single GRIB record from UNIX file-descriptor, NUNIT, unpack all header and data values into the appropriate arrays, and fill the allocatable array, DATARRAY(:).

Input: NUNIT (integer): C Unit to read from.

NDATA (integer): Size of array DATA (Should be \geq NDAT as computed herein.)

Output: DATA (real): The unpacked data array (dimension size of NDATA)
IERR(integer): Error flag, non-zero means there was a problem.

Side Effects: Header arrays SEC0, SEC1, SEC2, SEC3, SEC4, XEC4, INFOGRID and INFOGRID are filled.

The BITMAP array is filled.

The C unit pointer is advanced to the end of the GRIB record.

H.3 SUBROUTINE GRIBHEADER (IERR)

Unpack the header of a GRIB record

IERR non-zero means there was a problem unpacking the grib header

IERR (integer)

H.4 SUBROUTINE GRIBDATA (DATARRAY, NDAT)

- Read and unpack the data in a GRIB record into array, DATARRAY

Input: NDAT (integer): The size of the data array we expect to unpack.

Output: DATARRAY (real): The unpacked data from the GRIB record (dimension size of NDAT)

Side Effects: - STOP – if it cannot unpack the data.

H.5 SUBROUTINE GRIBPRINT (ISEC)

Print the header information from GRIB section ISEC.

ISEC Information can be found in the section H.10

H.6 SUBROUTINE GET_SEC1 (KSEC1)

- Return the header information from GRIB Section 1 (see H.10, TABLE 4).
- Return the GRIB Section 1 header information, which has already been unpacked by subroutine GRIBHEADER.
- KSEC1 (integer :: dimension(100))

H.7 SUBROUTINE GET_SEC2 (KSEC2)

- Return the header information from GRIB Section 2 (see H.10, TABLE 5).
- Return the GRIB Section 2 header information, which has already been unpacked by subroutine GRIBHEADER.
- KSEC2 (integer :: dimension(10))

H.8 SUBROUTINE GET_GRIDINFO (IGINFO, GINFO)

- Return the grid information of the previously-unpacked GRIB header.
 - IGINFO (integer :: dimension(40))
 - GINFO (real :: dimension(40))
- ** NOTE IGINFO and GINFO contain equivalent information, except that IGINFO is the integer form and GINFO is the real form.

H.9 C-ROUTINE COPEN (UNIT, NUNIT, NAME, MODE, ERR, OFLAG)

- Opens the GRIB file for reading later by the other subroutines

UNIT = Fortran unit number (integer)

NUNIT = UNIX file descriptor associated with 'unit' (integer)

NAME = UNIX file name (character (len=120))

MODE = 0 : write only - file will be created if it doesn't exist, - otherwise
will be rewritten (integer)
= 1 : read only

= 2 : read/write

ERR = 0 : no error opening file (integer)
!= 0 : Error opening file

OFLAG = 0 : file name printed (no errors printed) (integer)
> 0 : file name printed and errors are printed
< 0 : no print at all (not even errors)

H.10 SEC Header Array Information Tables

Please refer to <http://www.wmo.ch/web/www/WDM/Guides/Guide-binary-2.html> for additional GRIB1 header information

Table 5: SEC2: GRIB Header Section 2 information

Octet	GDS Content	
1-3	Length of GRIB Section 2 (in octets)	
4	Number of vertical-coordinate parameters	
5	Starting-point of the list of vertical-coordinate parameters	
6	Data-representation type (i.e., grid type) See GRIB Table 6 0 = Latitude/Longitude grid 3 = Lambert-conformal grid. 5 = Polar-stereographic grid.	
if(sec2(4)==0) then Lat/lon grid		
INFOGRID	Octet	GDS Content
1	7-8	I Dimension of the grid
2	9-10	J Dimension of the grid
3	11-13	Starting Latitude of the grid.
4	14-16	Starting Longitude of the grid.
5	17	Resolution and component flags.
6	18-20	Ending latitude of the grid.
7	21-23	Ending longitude of the grid.
8	24-25	Longitudinal increment.
9	26-27	Latitudinal increment.
10	28	Scanning mode (bit 3 from Table 8)
21	28	Iscan sign (+1/-1) (bit 1 from Table 8)
22	28	Jscan sign (+1/-1) (bit 2 from Table 8)
if(sec2(4)==1) then mercator grid		
INFOGRID	Octet	GDS Content
1	7-8	I Dimension of the grid

2	9-10	J Dimension of the grid
3	11-13	Starting Latitude of the grid.
4	14-16	Starting Longitude of the grid.
5	17	Resolution and component flags.
6	18-20	Ending latitude of the grid.
7	21-23	Ending longitude of the grid.
8	24-26	LATIN- The latitude(s) at which Mercator projection cylinder intersects the earth
9	27	Reserved (set to 0)
10	28	Scanning mode (bit 3 from Table 8)
11		True Lat
21	29-31	Iscan sign (+1/-1) (bit 1 from Table 8)
22	32-34	Jscan sign (+1/-1) (bit 2 from Table 8)
if(sec2(4)==3) then Lambert Conformal grid		
INFOGRID	Octet	GDS Content
1	7-8	I Dimension of the grid
2	9-10	J Dimension of the grid
3	11-13	Starting Latitude of the grid.
4	14-16	Starting Longitude of the grid.
5	17	Resolution and component flags.
6	18-20	Center longitude of the projection.
7	21-23	Grid-spacing in the I direction
8	24-26	Grid-spacing in the J direction
9	27	Projection center
10	28	Scanning mode (bit 3 from Table 8)
11	29-31	First TRUELAT value.
12	32-34	Second TRUELAT value.
13	35-37	Latitude of the southern pole
14	38-40	Longitude of the southern pole
21	41	Iscan sign (+1/-1) (bit 1 from Table 8)
22	42	Jscan sign (+1/-1) (bit 2 from Table 8)
if(sec2(4)==4) then Gaussian grid		
INFOGRID	Octet	GDS Content
1	7-8	I Dimension of the grid
2	9-10	J Dimension of the grid
3	11-13	Starting Latitude of the grid.
4	14-16	Starting Longitude of the grid.
5	17	Resolution and component flags.
6	18-20	Ending latitude of the grid.
7	21-23	Ending longitude of the grid.
8	24-25	Longitudinal increment.
9	26-27	Latitudinal increment.
10	28	Scanning mode (bit 3 from Table 8)
21	28	Iscan sign (+1/-1) (bit 1 from Table 8)

22	28	Jscan sign (+1/-1) (bit 2 from Table 8)
INFOGRID	if(sec2(4)==5) then Octet Polar stereographic grid	GDS Content
1	7-8	I Dimension of the grid
2	9-10	J Dimension of the grid
3	11-13	Starting Latitude of the grid.
4	14-16	Starting Longitude of the grid.
5	17	Resolution and component flags.
6	18-20	Center longitude of the projection.
7	21-23	Grid-spacing in the I direction
8	24-26	Grid-spacing in the J direction
9	27	Projection center
10	28	Scanning mode (bit 3 from Table 8)
21	29	Iscan sign (+1/-1) (bit 1 from Table 8)
22	30	Jscan sign (+1/-1) (bit 2 from Table 8)
INFOGRID	if(sec2(4)==50) then Spherical Harmonic Coefficients Octet	GDS Content
1	7-8	J-pentagonal resolution parameter
2	9-10	K-pentagonal resolution parameter
3	11-12	M-pentagonal resolution parameter
4	13	Spectral representation type (ON388 Table 9)
5	14	Coefficient storage mode (ON388 Table 10)
	15-32	Set to 0 (reserved)

H.11 Additional information for setting up the READ_GRIB routines for use on Linux Machines

A few steps were taken to modify the original READ_GRIB routines to make them more compatible with Absoft, Lahey95, and other Linux (32-bit and 64-bit) based compilers. Here is a list of those steps:

Replaced the extensions of each *.F file with *.F90.

In the C-routine, cio.c, the following lines of code were added or modified:

- Line 19 Added “|| defined(ABSOFT)and || defined(LAHEY)”

In the Makefile, the following lines of code were added or modified:

- Line 19 Added “.F90” to .SUFFIXES rule
- Line 34-35 – Added “@echo make absoft” and “@echo make lahey”; resp.
- Line 71 (and following lines) Added flags and compiler names for ABSOFT:

absoft:

```

$(MAKE) $(LIBTARGET) \
"FC = f90" \
"FCFLAGS = -O -YEXT_NAMES=LCS -B108 -YCFRL=1 -YDEALLOC=ALL - \
DHIDE_SHR_MSG -DNO_SHR_VMATH -DABSOFT -DLITTLE_ENDIAN -DBIT -DBIT32" \
"CC = gcc" \
"CCFLAGS = -O -Wall -DABSOFT -DLITTLE_ENDIAN -DG_ENABLE_DEBUG=1" \
"CPP = /lib/cpp" \
"CPPFLAGS = -C -P -DBIT32"

lahey:
$(MAKE) $(LIBTARGETS) \
"FC = lf95" \
"FCFLAGS = -O -DBIT32 -DLINUX -DLAHEY -DLITTLE_ENDIAN" \
"CC = cc" \
"CCFLAGS = -O -DUSE_GCC -DLAHEY -DLITTLE_ENDIAN" \
"CPP = /lib/cpp" \
"CPPFLAGS = -C -P "

```

- Line 90 Changed ".F.o" to ".F90.o"
- Line 91 Removed "-d" from the rule

H.12 Example of Fortran code that calls READ_GRIB routines

```

!-- Initialize certain variables and parameters of GRIB file:
nunit = 10           ! Fortran unit number
ufn = nunit + 1     ! UNIX file descriptor associated with "nunit"
datarray = 0

!-- Open INPUT GRIB File:
call copen (nunit, ufn, trim(input_file)//char(0), 1, iret, 1)
print *, " ** Open File Code: ", iret

if ( iret > 0) then      ! Return File Error Code Number - IF FAILED TO OPEN!
    write(*,*) "STOPPING ROUTINE -- FILE NOT OPENED DUE TO CODE # :: ", iret
    stop
end if

!-- Read GRIB file:
call gribread ( ufn, datarray, ndata, ierr )

if ( ierr > 0) then    ! Return File Error Code Number - IF FAILED TO READ!
    write(*,*) "STOPPING ROUTINE -- FILE NOT READ DUE TO CODE # :: ", ierr

```

```
        stop
end if

!Print GRIB Header Information
do isec = 0, 2
    call gribprint (isec)
end do

do i = 1, ndata
    if (datarray(i) >0 ) then
        print *, i, datarray(i)
    end if
end do
```

DRAFT

Table 3: SEC0: GRIB Header Section 0 information

Number	Description
1	Length of a complete GRIB record
2	Grib Edition Number

Table 4: SEC1: GRIB Header Section 1 information

Octet	PDS content
1-3	Length of GRIB section 1 (3 bytes)
4	Parameter Table Version number
5	Center Identifier
6	Generating process Identifier
7	Grid ID number for pre-specified grids.
8	Binary bitmap flag:
9	Parameter ID Number and Units (ON388 Table 2)
10	Indicator of level type or layer (ON388 Table 3)
11	Level value (height or pressure), of the top value of a layer
12	Level value, but for bottom value of a layer (0 if NA ??)
13	Year (00-99)
14	Month (01-12)
15	Day of the month (01-31)
16	Hour (00-23)
17	Minute (00-59)
18	Forecast time unit: (ON388 Table 4)
19	Time period 1 (Number of Time Units Given in Octet 18)
20	Time period 2 or time interval between successive analyses
21	Time range indicator (ON833 Table 5)
22-23	Number included in average when Octet 21 (Table 5) indicates average or accumulation (otherwise set to 0)
24	Number missing from averages or accumulations
25	Century (Years 1999 and 2000 are century 20, 2001 is century 21)
26	Sub-center identifier
27-28	Decimal scale factor D. Negative value indicates setting high order bit in Octet 27 to 1 ("on").
29	Is there a GDS (0=no, 1=yes; bit 1 of sec1(6)) Refer to Octet 8 above
30	Is there a BMS (0=no, 1=yes; bit 2 of sec1(6)) Refer to Octet 8 above

I GRIB Output Information

Introduction:

This is a package of subroutines to write GRIB data in LIS. As land surface models become more sophisticated, more variable metadata will need to be added. The package currently supports all of the mandatory ALMA output variables for LSMs. However, the GRIB output module can easily be extended to support additional variables. This new LIS grib interface was adapted from a module similair to one used in the AFWA AGRMET model. Supported LIS projections include Lat/Lon, Lambert Conformal, Polar Stereographic, and Mercator. The setup of the GRIB grid description section (GDS section) is handled automatically by LIS within the domain initialization module.

Charles J. Alonge
SAIC/NASA GSFC
Winter 2006/2007, and continuing

The user interfaces are:

SUBROUTINE GRIB1_SETUP - Initializes variable independent information in the GRIB product definition section (PDS).

SUBROUTINE GRIB1_FINALIZE - Finalizes the product definition section of the GRIB record by encoding variable specific metadata into the output grib record.

SUBROUTINE DRV_WRTIEVAR_GRIB - Writes the grib record (and stats) by gathering the variable from the individual MPI tasks (if applicable) and calls the lower level grib output routines

I.1 SUBROUTINE GRIB1_SETUP

- Call: GRIB1_SETUP(SECT1, INFO, BITMAP, DATE)
- Initialize variable independent information in the GRIB product definition record (e.g. center, subcenter, time valid, and bitmap flag)
- NOTE!: BITMAP MUST ALWAYS BE TRUE as the code will handle output over all gridpoints (including water) or just LIS output over land.

Input: SECT1 (integer(36)): GRIB PDS array. Contains variable specific GRIB metadata.

INFO (integer(5)): GRIB packing descriptor, describes the length of each section of grib header for low-level grib encoding.

BITMAP (logical) : Determines if a bitmap is used in the grib encoding (ALWAYS leave defined as true!!)

DATE (character(10)): Character string containing the date of the grib record (format: YYYYMMDDHH)

Side Effects: NONE - Do not set Bitmap to false or grib packing will fail.

I.2 SUBROUTINE GRIB1_FINALIZE

- Call: SUBROUTINE GRIB1_FINALIZE(GRIB_INDEX, SECT1, TIME_UNIT, TIME_1, TIME_2, TIME_RANGE)
- Finalizes the PDS section of a grib record by encoding variable specific metadata into the output grib record.

Input: GRIB_INDEX (integer): Index into the GRIB pds values array corresponding to the variable being processed for output (See tables below for enumerated types of this value for the different output variables)
SECT1 (integer (36)): Parameters for Section 1 (PDS) of the GRIB record
TIME_UNIT (integer): Units in time for output contained in GRIB record.
These are encoded as follows:

- 0** - Minute
- 1** - Hour
- 2** - Day
- 3** - Month
- 4** - Year
- 254** - Second

TIME_1(integer): Time1 for GRIB Record Descriptor. Used only when averaging or accumulating variables over a period of time (See TIME_RANGE description).

TIME_2(integer): Time2 for GRIB Record Descriptor. For general LIS usage this should always be set to zero.

TIME_RANGE: Time Range Indicator. Describes relationship between TIME_1 and TIME_2. These are encoded as follows (for a more detailed description please refer to the WMO Grib 1 Manual - Table 5):

- 0** - Forecast product value for reference TIME_1 (TIME_2 ignored)
- 1** - Analysis product for reference TIME_1 (TIME_2 ignored)
- 2** - Product wth a valid time ranging between +TIME_1 and +TIME_2
- 3** - Average (reference time +TIME_1 to reference time +TIME_2)
- 4** - Accumulation (reference time +TIME_1 to reference time +TIME_2)
product considered valid at TIME_2
- 5** - Difference (reference time +TIME_2 minus reference time +TIME_1)
product considered valid at TIME_2
- 6** - Average (reference time -TIME_1 to reference time -TIME_2)
- 7** - Average (reference time -TIME_1 to reference time +TIME_2)

Side Effects: Do not use negative values when defining the three time descriptor variables as this will cause an error in the low-level grib packing routines. Instead use a different time range indicator.

I.3 SUBROUTINE DRV_WRITEVAR_GRIB

- Call: DRV_WRITEVAR_GRIB(FTN, FTN_STATS, N, FLAG, VAR, MVAR, FORM, TOPLEV, BOTLEV, KDIM)
- Writes the grib record by gathering the variable from the individual MPI tasks and calling the lower level grib output routines

Input: FTN (integer): Unit number of grib output file.
 FTN_STATS (integer): Unit number of grib output file.
 N (integer): Index of the LIS domain or nest.
 FLAG (integer): Unit number of grib output file.
 VAR (real(lis%nch(n))): Variable output data.
 MVAR (character(len=*)): Name of variable being written.
 FORM (integer): Format to be used in stats file (1-decimal,2-scientific)
 TOPLEV (real(KDIM)): Format to be used in stats file (1-decimal,2-scientific)
 BOTLEV (real(KDIM)): Format to be used in stats file (1-decimal,2-scientific)
 KDIM (integer): Grid dimension Format to be used in stats file (1-decimal,2-scientific)
 INFO (integer(5)): GRIB packing descriptor, describes the length of each section of grib header for low-level grib encoding.
 BITMAP (logical) : Determines if a bitmap is used in the grib encoding (ALWAYS leave defined as true!!)
 DATE (character(10)): Character string containing the date of the grib record (format: YYYYMMDDHH)

Side Effects: NONE - Do not set Bitmap to false or grib packing will fail.

I.4 Output GRIB Variables

GRIB ID	Table Number	Description
ALMA ENERGY BALANCE COMPONENTS		
GRIB_SWNET	1	Net Shortwave Radiation Flux
GRIB_LWNET	2	Net Longwave Radiation Flux
GRIB_QLE	3	Latent Heat Flux
GRIB_QH	4	Sensible Heat Flux
GRIB_QG	5	Ground Heat Flux
GRIB_QF	6	Energy of Fusion
GRIB_QV	7	Energy of Sublimation
GRIB_QTAU	8	Momentum Flux
GRIB_QA	9	Advection Energy
GRIB_DELSRFHEAT	10	Change in Surface Heat Storage
GRIB_DELCLDCNT	11	Change in Snow Cold Content
ALMA WATER BALANCE COMPONENTS		

GRIB_SNOWF	12	Snowfall Rate
GRIB_RAINF	13	Rainfall Rate
GRIB_EVAP	14	Total Evaporation
GRIB_QS	15	Surface Runoff
GRIB_QREC	16	Recharge
GRIB_QSB	17	Subsurface Runoff
GRIB_QSM	18	Snowmelt
GRIB_QFZ	19	Refreezing of Water in the Snow
GRIB_QST	20	Snow Throughfall
GRIB_DELSM	21	Change in Soil Moisture
GRIB_DELSWE	22	Change in Snow Water Equivalent
GRIB_DELSRFSTR	23	Change in Surface Water Storage
GRIB_DELINTCPT	24	Change in Interception Storage
ALMA SURFACE STATE VARIABLES		
GRIB_SNOWT	25	Snow Surface Temperature
GRIB_VEGT	26	Vegetation Canopy Temperature
GRIB_BARESOILT	27	Temperature of Bare Soil
GRIB_AVGSURFT	28	Average Surface Temperature
GRIB_RADT	29	Surface Radiative Temperature
GRIB_ALBEDO	30	Surface Albedo
GRIB_SWE	31	Snow Water Equivalent (SWE)
GRIB_SWEVEG	32	SWE intercepted by Vegetation
GRIB_SURFSTOR	33	Surface Water Storage
ALMA SUBSURFACE STATE VARIABLES		
GRIB_SOILMOIST	34	Soil Moisture
GRIB_SOILTEMP	35	Soil Temperature
GRIB_LSOILMOIST	36	Avg. layer fraction of Liquid Moisture
GRIB_FSOILMOIST	37	Avg. layer fraction of Frozen Moisture
GRIB_SOILWET	38	Total Soil Wetness
ALMA EVAPORATION COMPONENTS		
GRIB_POTEVAP	39	Potential Evaporation
GRIB_ECANOP	40	Interception Evaporation
GRIB_TVEG	41	Vegetation Transpiration
GRIB_ESOIL	42	Bare Soil Evaporation
GRIB_EWATER	43	Open Water Evaporation
GRIB_ROOTMOIST	44	Root Zone Soil Moisture
GRIB_CANOPINT	45	Total Canopy Water Storage
GRIB_ESNOW	46	Snow Evaporation
GRIB_SUBSNOW	47	Snow Sublimation
GRIB_SUBSURF	48	Sublimation of Snow Free Area
GRIB_ACOND	49	Aerodynamic Conductance
FORCING VARIABLES		
GRIB_WINDFORC	50	Wind Speed
GRIB_RAINFFORC	51	Rainfall Forcing
GRIB_SNOWFFORC	52	Snowfall Forcing

GRIB_TAIRFORC	53	Air Temperature
GRIB_QAIRFORC	54	Specific Humidity
GRIB_PSLANDFORC	55	Surface Pressure
GRIB_SWDOWNFORC	56	Downwelling Shortwave Radiation Flux
GRIB_LWDOWNFORC	57	Downwelling Longwave Radiation Flux
PARAMETER OUTPUT - EXPERIMENTAL		
GRIB_LANDMASK	58	Land Mask
GRIB_LANDCOVER	59	Vegetation Type - Landcover
GRIB_SOILTYPE	60	Soil Type
GRIB_SOILCOLOR	61	Soil Color
GRIB_TOPOGRAPHY	62	Topography of Land Surface
GRIB_LAI	63	Leaf Area Index
GRIB_SAI	64	Stem Area Index
GRIB_SNFRALBEDO	65	Snow-free Albedo
GRIB_MXSNALBEDO	66	Maximum Snow Albedo
GRIB_GREENNESS	67	Greenness Fraction
GRIB_TEMP_BOT	68	Bottom Temperature

I.5 Example of Fortran code that calls GRIB output routines

```

! Setup of GRIB Metadata Section

! toplev is the depth of the top of each soil layer
! botlev is the depth of the bottom of each soil layer
toplev(1) = 0.0
botlev(1) = noah_struc(n)%lyrthk(1)

! determine bounding levels for each soil moisture layer
do i = 2, noah_struc(n)%nslay
    toplev(i) = toplev(i-1) + noah_struc(n)%lyrthk(i-1)
    botlev(i) = botlev(i-1) + noah_struc(n)%lyrthk(i)
enddo

! Convert to centimeters -- the depths for layers below the land
! surface (surface = 112) are expected in centimeters,
! per GRIB specifications.
toplev = toplev * 100.0
botlev = botlev * 100.0

! Set values for non layered fields (Fluxes, Sfc Fields, etc.)
toplev0 = 0
botlev0 = 0

! Set Date String to Pass to GRIB Module

```

```

hr1=lis%hr
da1=lis%da
mo1=lis%mo
yr1=lis%yr
write(unit=date,fmt='(i4.4,i2.2,i2.2,i2.2)') yr1,mo1,da1,hr1

! Setup common information to go into the PDS section
! of the GRIB file (Variable Independent Metadata)

call grib1_setup(gribobj(n)%sect1, gribobj(n)%info, &
                  gribobj(n)%bitmap, date)

! Set time units of output for GRIB file
! Is output interval in days,hours,minutes,seconds?
! Also, determine output time range in that unit (time_past)

if(noah_struc(n)%outInterval .GT. 0) then
    time_unit = 254      ! seconds
    time_past = (noah_struc(n)%outInterval / 1)
endif
if(noah_struc(n)%outInterval .GE. 60) then
    time_unit = 0        ! minutes
    time_past = (noah_struc(n)%outInterval / 60)
endif
if(noah_struc(n)%outInterval .GE. 3600) then
    time_unit = 1        ! hours
    time_past = (noah_struc(n)%outInterval / 3600)
endif
if(noah_struc(n)%outInterval .GE. 86400) then
    time_unit = 2        ! days
    time_past = (noah_struc(n)%outInterval / 86400)
endif

! End of GRIB metadata section

! Sample Outputs.....
```

~~SECRET~~

```

! Output Net Shortwave Radiation

! Finalize PDS section of this variable, in this case SWNET is a time
! averaged variable => time range indicator equals 7
call grib1_finalize(GRIB\_SWNET,gribobj(n)%sect1,time_unit,time_past,0,7)

noah_struc(n)%noah%swnet = noah_struc(n)%noah%swnet/float(noah_struc(n)%count)

! Call low level grib routines (single level data, hence 1 as final arg)
```

```

call drv_writevar_grib(ftn,ftn_stats,n,metadata_output%swnet,      &
noah_struc(n)%noah%swnet,"Swnet(W/m2)",1, &
toplev0, botlev0, 1)

! Output Soil Moisture

! Store soil moisture in a 2D array (LSM points x Num. Layers)
temp_nslay = 0.0
do i=1,noah_struc(n)%nslay
  do t=1,lis%nch(n)
    temp_nslay(t,i) = noah_struc(n)%noah(t)%soilmoist(i)
  enddo
enddo

! Finalize PDS section of this variable, in this case SOILMOIST is an
! instantaneous variable => time range indicator equals 1 - analysis
call grib1_finalize(GRIB\_SOILMOIST,gribobj(n)%sect1,time_unit,0,0,1)

! Call low level grib routines (output all levels, number of layers
! is used as the last argument)
call drv_writevar_grib(ftn,ftn_stats,n,metadata_output%soilmoist,&
temp_nslay,"SoilMoist(kg/m2)",2,           &
toplev, botlev, noah_struc(n)%nslay)

```

References

- Berg, A., J. Famiglietti, J. Walker, and P. Houser, 2003: Impact of bias correction to reanalysis products on simulations of north american soil moisture and hydrological fluxes. *Journal of Geophysical Research*, **4490**, doi:10.1029/2002JD003334.
- Sawyer, W. and A. da Silva, 1997: Protex: A sample fortran 90 source code documentation system. Technical report, NASA GMAO, dAO Office Note 97-11.
- Sheffield, J., G. Goteti, and E. Wood, 2006: Development of a 50-yr high-resolution global dataset of meteorological forcings for land surface modeling. *Journal of Climate*, **13**, 3088–3111.